# EX-94421/A

# Analog Input and
Multi-Function Digital I/O Card

## Software Manual (V1.0)

旭蒙科技股份有限公司

**TOPSCCC TECHNOLOGY CO., LTD.**

台北市內湖區陽光街 345 巷 12 號 5 樓
5F., No.12, Lane 345, Yang Guang St.
Neihu, 114 Taipei, Taiwan
TEL：+886-2-2799-9080
FAX：+886-2-2658-5042
http://www.topsccc.com.tw
E-mail：tops@topsccc.com

# Correction record

| Version | Record |
|---|---|
| 1.0 | |

# Contents

# 1. How to install the software of EX-94421A

## 1.1 Install the PCI driver

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

In Windows 2000/XP/Win7 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 (..\EX94421A\software\Win2K_up\driver or if you download from website please execute the self-unzip file EX94421A_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

## 1.2 Install the Linux driver

--To unpack the file, use the "tar" command, Use the syntax :

```
$tar -zxvf <filename>
```

This creates the EX94421A directory, containing installation scripts,
Makefile, driver source , dynamic library, demo source and executing.

--Change to <filename> directory.

```
$cd <filename>
```

--First install, type "make" to compile the source.

```
$make clean
$make
```

You may see the drv94421A.ko module file.

--We will install modules, Dynamic library, Demo and shell script on boot, Use the script command :

```
$./install94421A setup
```

--Then load modules and create device :

```
$./install94421A start
```

--Executing demo, You can use the following commands under any directory.

```
$EX94421Ademo
```

--Want to uninstall EX94421A driver , can use the command :

```
$./install94421A uninstall
```

Note : if you change CardID,please type "./install94421A reload" command first.
Note : if you executing demo and returns error,maybe you have not install QT or KDE library.

Please install QT or KDE by the following instruction:

```
yum install qt*(kde*)
```

# 2. About the EX-94421A software

EX-94421A 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 EX-94421A 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 EX-94421A functions within Windows' operation system environment.

## 2.1 What you need to get started

To set up and use your EX-94421A software, you need the following:

- EX-94421A software
- EX-94421A hardware

Main board

Wiring board (Option)

## 2.2 Software programming choices

You have several options to choose from when you are programming EX-94421A 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 EX-94421A software.

# 3. EX-94421A Language support

The EX-94421A software library is a DLL used with Windows 2000/XP/Win7. You can use these DLL with any Windows integrating development environment that can call Windows DLLs.

## 3.1 Building applications with the EX-94421A software library

The EX94421A function reference topic contains general information about building EX-94421A applications, describes the nature of the EX-94421A files used in building EX-94421A 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 EX-94421A Windows libraries

The EX-94421A for Windows function library is a DLL called **EX94421A.dll**. Since a DLL is used, EX-94421A functions are not linked into the executable files of applications. Only the information about the EX-94421A functions in the EX-94421A 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 EX-94421A functions in EX94421A.dll.

| **Header Files and Import Libraries for Different Development Environments** | | |
|---|---|---|
| Development Environment | Header File | Import Library |
| Microsoft C/C++ | EX94421A.h | EX94421Avc.lib |
| Borland C/C++ | EX94421A.h | EX94421Abc.lib |
| Microsoft Visual Basic | EX94421A.bas | |

**Table 1**

# 4. Function format and language difference

## 4.1 Function format

Every EX-94421A 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 EX-94421A function is the parameter **CardID** which is located the driver of EX94421A board you want to use those given operation. The **CardID** is assigned by DIP/ROTARY 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 DIP/ROTARY SW (**0x0-0xF**)

## 4.2 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**

## 4.3 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 EX94421A API. Read the following sections that apply to your programming language.

**Note:** Be sure to include the declaration functions of EX-94421A prototypes by including the appropriate EX-94421A header file in your source code. Refer to Building Applications with the EX-94421A Software Library for the header file appropriate to your compiler.

### 4.3.1 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 AD function has the following format:

```
Status = EX94421A_AD_data_read(CardID, channel, *voltage_data);
```

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

```
    u8 CardID, channel;
    i16 voltage_data,
    u32 Status;
    Status = EX94421A_AD_data_read (CardID, channel, voltage_data);
```

### 4.3.2 Visual basic

The file EX94421A.bas contains definitions for constants required for obtaining AIO Card information and declared functions and variable as global variables. You should use these constants symbols in the EX94421A.bas, do not use the numerical values.

In Visual Basic, you can add the entire EX94421A.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 EX94421A.bas file for your project in Visual Basic 4.0, go to the **File** menu and select the **Add File... option**. Select EX94421A.bas, which is browsed in the EX94421A \ api directory. Then, select **Open** to add the file to the project.

To add the EX94421A.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** EX94421A.bas, which is in the EX94421A \ api directory. Then, select **Open** to add the file to the project.

### 4.3.3 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 EX94421Abc.lib EX94421A.dll

Then add the **EX94421Abc.lib** to your project and add

#include “EX94421A.h” to main program.

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

```
Status =    EX94421A_AD_data_read(CardID, channel, *voltage_data);
```

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

```
u8 CardID, channel;
i16 voltage_data,
u32 Status;
Status =EX94421A_AD_data_read (CardID, channel, voltage_data);
```

# 5. Software overview and dll function

These topics describe the features and functionality of the EX-94421A boards and briefly describes the EX-94421A functions.

## 5.1 Initialization and close

You need to initialize system resource each time you start to run your application.

***EX94421A_initial( )*** will do.

Once you want to close your application, call

***EX94421A_close( )*** to release all the resource.

If you want to know the physical address assigned by OS. use

***EX94421A_info( )*** to get the address.

- **EX-94421A_initial**

**Format :** **u32 Status =EX94421A_initial (void)**

**Purpose:** Initial the EX-94421A resource when start the Windows applications.

- **EX-94421A_close**

**Format :** **u32 Status =EX94421A_close (void);**

**Purpose:** Release the EX-94421A resource when close the Windows applications.

- **EX-94421A_info**

**Format :** **u32 status =EX94421A_info(u8 CardID,u16 *address)**

**Purpose:** Read the physical I/O address assigned by O.S..

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |

**Output:**

| Name | Type | Description |
|---|---|---|
| address | u16 | physical I/O address assigned by OS |

## 5.2 Analog input

The EX-94421A now is a 16 bit AD cards. You must configure the input range of the specific channel by:

***EX94421A_AD_config_set( )*** and read back the configuration for verification by:

***EX94421A_AD_config_read( )***

To read the input voltage value by:

***EX94421A_AD_value_read( ),*** it can be also read data by

***EX94421A_AD_data_read( )***

The EX-94421A hardware only provide the AD conversion data on the fly, in noisy environment the conversion result maybe contaminated by noise, to use the integral of signals will eliminate the high frequency noise. The dll has provide build in software integration functions; to start the function by:

***EX94421A_AD_integral_start( )*** and read the integration data by

***EX94421A_AD_integral_all_read( ),*** if you want to stop the integration function don't forget to release the resource and stop integration by:

***EX94421A_AD_integral_stop( )***

- **EX-94421A_AD_config_set**

**Format :** **u32 status = EX94421A_AD_config_set(u8 CardID,u8 channel,u8 mode)**

**Purpose:** Set A/D config.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| channel | u8 | A/D channel number<br>0~7: EX94421A, 8 channels AD |
| mode | u8 | scale range:<br>0: 0V ~ 5V<br>1: -5V ~ +5V<br>2: 0V ~ 10V<br>3: -10V ~ +10V<br>255 : AD stop operation. |

- **EX-94421A_AD_config_read**

**Format :** **u32 status = EX94421A_AD_config_read(u8 CardID,u8 channel,u8 *mode)**

**Purpose:** Read A/D configuration.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| channel | u8 | A/D channel number<br>0~7: EX-94421A, 8 channels AD |

**Output:**

| Name | Type | Description |
|---|---|---|
| mode | u8 | scale range:<br>0: 0V ~ 5V<br>1: -5V ~ +5V (Default)<br>2: 0V ~ 10V<br>3: -10V ~ +10V<br>255 : AD stop operation. |

- **EX-94421A_AD_value_read**

**Format :** **u32 status = EX94421A_AD_value_read(u8 CardID,u8 channel, f32 *voltage_value)**

**Purpose:** Read voltage value with pre-calibration data.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| channel | u8 | A/D channel number<br>0~7: EX-94421A, 8 channels AD |

**Output:**

| Name | Type | Description |
|---|---|---|
| voltage_value | f32 | Voltage value based on the AD converted and calibrated data.<br>Say if the AD scale range is set at 0~5V then the voltage value returned will be in the 0~5 range. |

- **EX-94421A_AD_data_read**

**Format :** **u32 status = EX94421A_AD_data_read(u8 CardID,u8 channel,**
**u16 *voltage_data)**

**Purpose:** Read voltage value with pre-calibration data.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| channel | u8 | A/D channel number<br>0~7: EX-94421A, 8 channels AD |

**Output:**

| Name | Type | Description |
|---|---|---|
| voltage_data | u16 | Voltage value based on the AD converted data.<br>If unipolar voltage: 0~5V or 0~10V is 0~65535<br>If bipolar voltage, please take the data as 2's complement, change to i16 first then -5V~+5V or -10V~+10V is -32768 ~ +32767 |

- **EX-94421A_AD_integral_start**

**Format :** **u32 status = EX94421A_AD_integral_start(u8 CardID,u8 mode)**

**Purpose:** start AD conversion with integral constant.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by jumper setting |
| mode | u8 | 0: immediately access, no integration<br>1: integration time 100ms<br>2: integration time 200ms<br>3: integration time 300ms<br>4: integration time 400ms<br>5: integration time 500ms<br>6: integration time 600ms<br>7: integration time 700ms<br>8: integration time 800ms<br>9: integration time 900ms<br>10: integration time 1s |

- **EX-94421A_AD_integral_all_read**

**Format :** **u32 status = EX94421A_AD_integral_all_read(u8 CardID,i16 data[8])**

**Purpose:** read one port integral result of AD conversion data.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by jumper setting |

**Output:**

| Name | Type | Description |
|---|---|---|
| data[8] | u16 | data[0]: Channel 0 AD data<br>…<br>data[7]: Channel 7AD data<br>If unipolar voltage: 0~5V or 0~10V is 0~65535<br>If bipolar voltage, please take the data as 2’s complement, change to i16 first then -5V~+5V or -10V~+10V is -32768 ~ +32767 |

**Note:**

To read all channels in integral

Start integral mode by EX94421A_AD_integral_start.

Read all channels by **EX94421A_AD_integral**_all_**read**.

Stop AD integration function by **EX94421A_AD_integral_stop.**

- **EX-94421A_AD_integral_stop**

**Format :** **u32 status = EX94421A_AD_integral_stop(u8 CardID)**

**Purpose:** stop AD integral conversion.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by jumper setting |

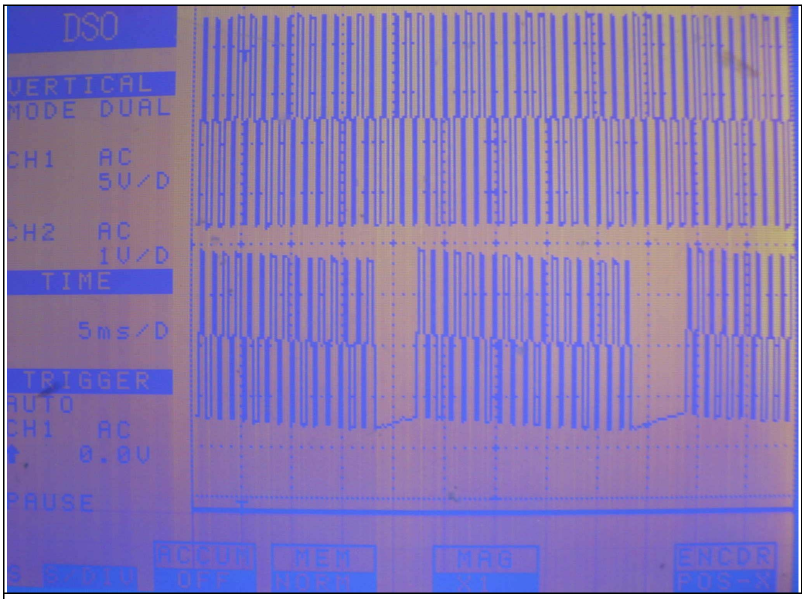
### 5.3 TTL I/O Port R/W

In general, TTL I/O port can be input or output as configured. To work as input, the EX-94421A series cards provide input digital debounce function.

Debounce is the function to filter the input jitters. From the microscope view of a switch input, you will see the contact does not come to close or release to open clearly. In most cases, it will contact-release-contact-release… for many times then go to steady state (ON or OFF). If you do not have the debounce function, you will read the input at high state and then next read will get low state, this maybe an error data for your decision of contact input.

Debounce can be implemented by hardware or software. Analog hardware debounce circuit will have fixed time constant to filter out the significant input signal, if you want to change the response time; the only way is to change the circuit device.

If digital debounce is implemented, maybe several filter frequency you can choose. To choose the filter frequency, please keep the Nyquist–Shannon sampling theorem in mind: **filter sample frequency must at least twice of the input frequency.** The following sample is a bad selection of debounce filter, the input frequency is not as low as les than half of the sample frequency and the output will generate a beat frequency.



<- Input frequency at 835Hz

<- Output of digital filter,
Please note the beat frequency.

Digital debounce circuit work at 1KHZ sample rate and observe the output of filter from 835Hz input.

Software debounce will consumes the CPU time a lot, we do not recommend to use except for you really know you want.

To configure the port as input or output by:

***EX94421A_TTL_IO_config_set ( )*** and read back the configuration by:

***EX94421A_TTL_IO_config_read ( ).***

The TTL I/O port can use:

***EX94421A_TTL_IO_port_set ( )*** to output data and input data by:

***EX94421A_TTL_IO_port_read( ).***

For the point output, use:

***EX94421A_TTL_IO_point_set ( )*** and point input by:

***EX94421A_TTL_IO_point_read ( )***.

At noisy environment to debounce the signal or to debounce the mechanical contact input, use:

***EX94421A_TTL_IO_debounce_time_set ( )*** to set the adequate time constant to drop out the noise and read back to check the setting by:

***EX94421A_TTL_IO_debounce_time_read( ).***

- **EX-94421A_TTL_IO_config_set**

**Format : u32 status =EX94421A_TTL_IO_config_set (u8 CardID, u8 port,**
**u8 configuration)**

**Purpose:** Sets port configuration.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |
| configuration | u8 | 0: input port (default)<br>1: output port |

- **EX-94421A_TTL_IO_config_read**

**Format : u32 status =EX94421A_TTL_IO_config_read (u8 CardID, u8 port,**
**u8 *configuration)**

**Purpose:** read port configuration.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |

**Output:**

| Name | Type | Description |
|---|---|---|
| configuration | u8 | 0: input port (default)<br>1: output port |

- **EX-94421A_TTL_IO_port_set**

**Format :** **u32 status = EX94421A_TTL_IO_port_set (u8 CardID,u8 port, u8 data)**

**Purpose:** Sets the port data.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |
| data | u8 | bitmap of output values<br>bit0: DIOn0<br>…<br>bit7: DIOn7 |

- **EX-94421A_TTL_IO_port_read**

**Format :** **u32 status = EX94421A_TTL_IO_port_read (u8 CardID , u8 port , u8 *data)**

**Purpose:** Read the port data.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |

**Output:**

| Name | Type | Description |
|---|---|---|
| data | u8 | bitmap of output/input values<br>bit0: DIOn0<br>…<br>bit7: DIOn7 |

- **EX-94421A_TTL_IO_point_set**

**Format :** **u32 status =EX94421A_TTL_IO_point_set (u8 CardID, u8 port, u8 point, u8 state)**

**Purpose:** Sets the bit data of output port.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |
| point | u8 | point number<br>0~7 for bit0~bit7 (DIOn0 ~ DIOn7) |
| state | u8 | point of output state |

- **EX-94421A_TTL_IO_point_read**

**Format :** **u32 status = EX94421A_TTL_IO_point_read (u8 CardID, u8 port, u8 point, u8 *state)**

**Purpose:** Read the output port state .

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |
| point | u8 | point number of input<br>0~7 for bit0~bit7 (DIOn0 ~ DIOn7) |

**Output:**

| Name | Type | Description |
|---|---|---|
| state | u8 | point of output/input state |

- **EX-94421A_TTL_IO_debounce_time_set**

**Format :** **u32 status = EX94421A_TTL_IO_debounce_time_set (u8 CardID,u8 port , u8 debounce_time)**

**Purpose:** debounce time of the TTL I/O port signal

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |
| debounce_time | u8 | Debounce time selection:<br>0: no debounce<br>1: filter out duration less than 10ms (100Hz, default)<br>2: filter out duration less than 5ms (200Hz)<br>3: filter out duration less than 1ms (1KHZ) |

**Note:**

only valid for port configured as input

- **EX-94421A_TTL_IO_debounce_time_read**

**Format :** **u32 status = EX94421A_TTL_IO_debounce_time_read (u8 CardID,u8 port , u8 *debounce_time)**

**Purpose:** To read back configuration of debounce mode

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| port | u8 | port number<br>0: port0 (DIO0x)<br>1: port1 (DIO1x) |

**Output:**

| Name | Type | Description |
|---|---|---|
| debounce_time | u8 | Debounce time selection:<br>0: no debounce<br>1: filter out duration less than 10ms (default)<br>2: filter out duration less than 5ms<br>3: filter out duration less than 1ms |

## 5.4 Counter / Timer / PWM function

Many control applications need timer as time base for digital sampled data control systems. The timer consists a counter to count the time base clock on the fly and generate interrupt on a periodic time interval. If the counter do not count the time base but count the signals from external world, we call it “counter”.

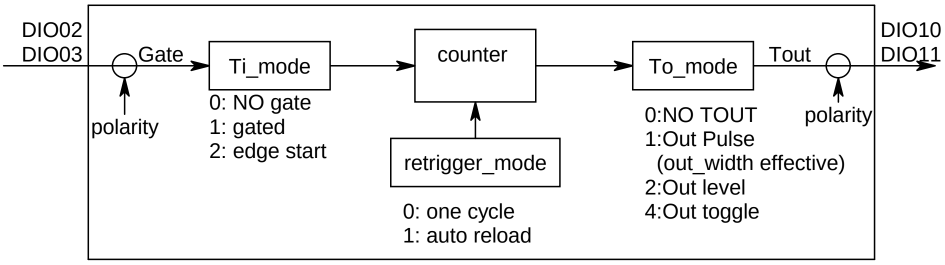
A timer/counter may be multi-functions, if the input signal and control mode and the output can be programmed as various kind of working mode.

### Input signal debounce

The timer / counter input comes from DIO00 ~ DIO03 the signal maybe occasionally contaminated by noise. EX-94421A series card provides wide range of filter frequency from 100Hz up to 1KHz (to drop out noise pulse less than 1ms), even the quadrature signals comes from mechanical contacts the counter can still operate very nice. If you will use faster signals, you can program the debounce as **no debounce** to pass the signal directly to counter. But take care of the noise induced by the environment and wiring, we recommended you to use a high speed isolation type encoder counter card such as LSI3101 (up to 8M counter speed) or LSI3101A (up to 16M counter speed) of TOPSCCC Technology Co Ltd to get better performance.

### Timer function

The timer model used in EX-94421A series is as follows:



In this model, the timer can work in one cycle mode and auto reload mode.

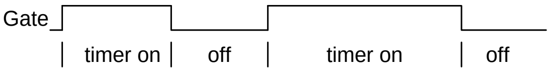
**one cycle mode**: the timer will stop when the timer time up.

**auto reload mode** ( sometimes called continuous mode): the time will reload the time constant while time up.

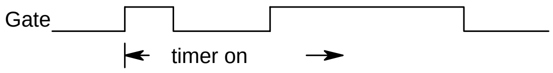
In the timer input control block:

**NO gate**: the timer do not control by any input.

**gated** : the timer only working on the gate input active and stops counting while gate is inactive.



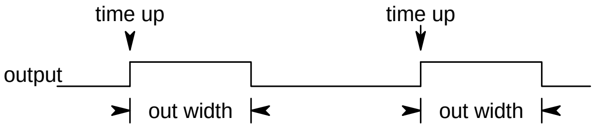
**Edge start**: the timer will start timing while the gate input transition from inactive to active.



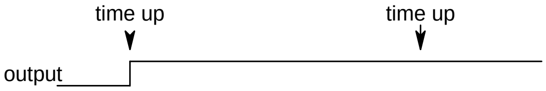
In the timer output block:

**NO TOUT**: the timer has no output to control (but timer time up interrupt is available).

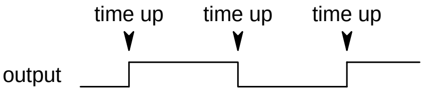
**Out_pulse:** while the timer time up, it will trigger an output pulse and pulse width is controlled by out_width register at 1us time base.



**Out_level:** while the timer time up, it will trigger the output active.



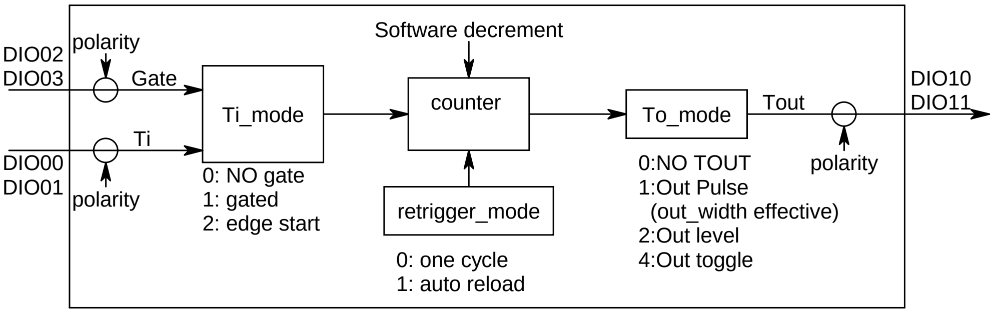
**Out_toggle:** while the timer time up, it will trigger the output toggled.



**polarity:** set the input/output active high or active low

## Counter function

The counter model used in EX94421A series is as follows:



In this model, the counter can work in one cycle mode and auto reload mode.

**one cycle mode**: the counter will stop when the counter cross zero.

**auto reload mode** ( sometimes called continuous mode): the counter will reload the counter constant while time up.

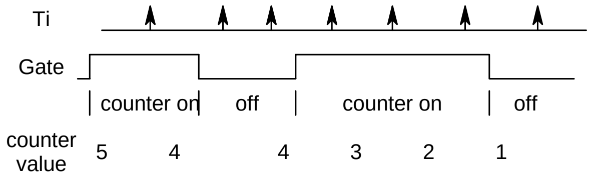
**Software decrement:** the counter value will decrement by software trigger.

In the counter input control block:

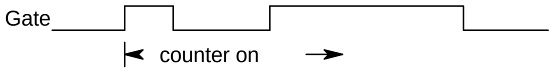
**NO gate**: the counter do not control by any input.

**gated** : while gate input is active and the counter signal input also active the counter will decrement by 1 and stops counting while gate is inactive.

Take the following diagram as example, the counter is initialized at 5 and working in gated mode, while the Ti (counter signal input) is active and gate is also active, the counter will decrease by one.



**Edge start**: the counter will start counting function while the gate input transition from inactive to active.



In the counter output block: (refer the timer function)

**NO TOUT**: the counter has no output to control (but counter cross zero interrupt is available).

**Out_pulse:** while the counter cross zero, it will trigger a output pulse and pulse width is controlled by out_width register at 1us time base.

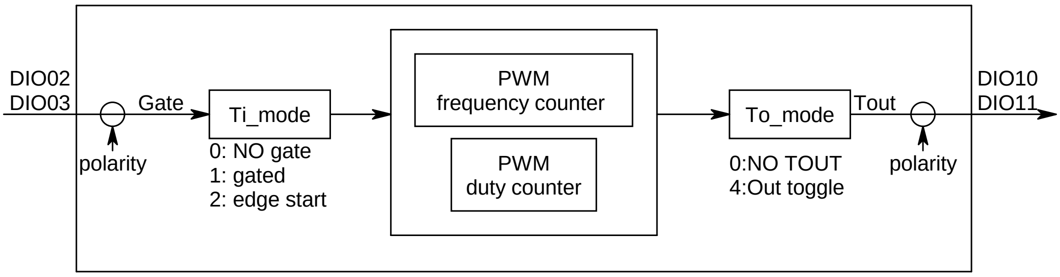
**Out_level:** while the counter cross zero, it will trigger the output active.

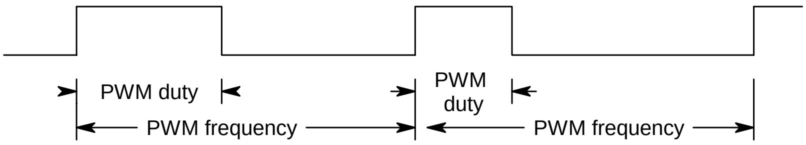
**Out_toggle:** while the counter cross zero, it will trigger the output toggled.

**polarity:** set the input/output active high or active low

## PWM function

The PWM model used in EX-94421A series is as follows:





In this model, the PWM counter can only work in auto reload mode.

**auto reload mode** ( sometimes called continuous mode): the time will reload the time constant while time up.

In the PWM counter input control block: (refer the counter function)

**NO gate**: the PWM counter do not control by any input.

**gated** : while gate input is active the PWM counter will start working and stops while gate is inactive.

**Edge start**: the PWM counter will start counting function while the gate input transition from inactive to active.

In the PWM counter output block: (refer the timer function)

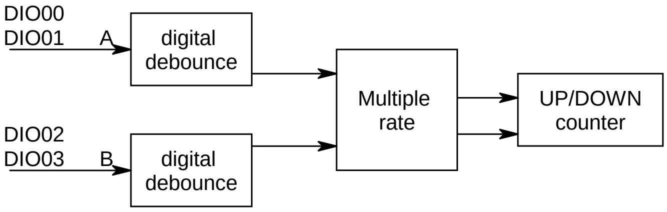
**NO TOUT**: the PWM counter has no output to control.

**Out_toggle:** while the PWM counter cross zero, it will trigger the output toggled.

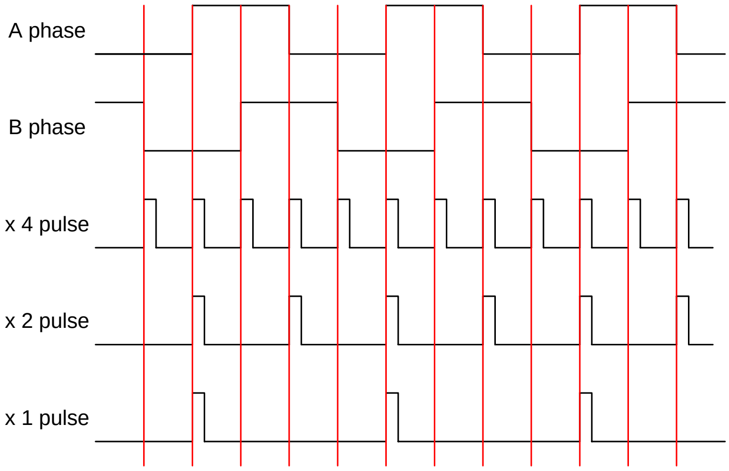
**polarity:** set the input/output active high or active low

### quadrature encoder counter

In spite of the flexible multi-function timer/counter, the quadrature encoder counter is another type of application. The EX-94421A series also has the build in function for quadrature encoder input counting.



On the above diagram, you can see the digital debounce function filter out the unwanted high frequency then pass the signal to the multiple rate circuit to determine the pulse and direction, finally the counter counts the pulses.



The left diagram shown that A phase leads B, if we take A leads B as up count and the counting pulse of up count will depends on the multiple rate.

**DLL functions of timer / counter**

Timer/counter function can work in general mode: as timer, as counter or as PWM generator and in special mode: quadrature counter mode.

In the general timer/counter mode, DIO00, DIO02 and DIO10 can be configured as dedicated I/O for timer1 / counter1 and DIO01, DIO03 and DIO11 can be configured as dedicated I/O for timer2 / counter2.

To configure the working mode use

***EX94421A_timer_set( )*** to configure as timer and its output mode

***EX94421A_counter_set( )*** to configure as counter and its input and output mode

***EX94421A_PWM_set( )*** to configure as PWM generator.

***EX94421A_quadrature_set( )*** to configure as quadrature counter.

To start/stop the operation by:

***EX94421A_TC_start( )***

***EX94421A_TC_stop( )***

To read or load dedicated timer/counter registers, use

***EX94421A_TC_set( )*** set TC dedicated registers

***EX94421A_TC_read( )*** read TC dedicated registers

If you want to change the input polarity, using

***EX94421A_TC_input_polarity_set( )*** and read back to verify by:

***EX94421A_TC_input_polarity_read( )***

If you want to change the output polarity, using

***EX94421A_TC_output_polarity_set( )*** and read back to verify by:

***EX94421A_TC_output_polarity_read( )***

## EX-94421A_timer_set

**Format :** **u32 status = EX94421A_timer_set (u8 CardID, u8 TimerID,**
**Timer_struct *TC_struct)**

**Purpose:** To setup timer operation mode or update timer

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| TC_struct | struct | struct TC_struct<br>{<br>u8 TiGate_MODE,<br>// 0: NO_GATE<br>//Always count without gate function,<br>//DIO02 / DIO03 is digital input.<br>// 1:GATED<br>//DIO02 / DIO03 is gate input,<br>//after command start_TC,<br>//if internal logic active high will start timer and //low will halt the timer counting.<br>// 2: EDGE_START<br>//DIO02 / DIO03 is gate input,<br>//after command start_TC,<br>//if internal logic active high will start timer<br>u32 time_constant,<br>// Timer constant based on 1us clock<br>u8 Tout_mode,<br>// 0: NO_TOUT ,<br>// DIO10 / DIO11: use as general digital output<br>// 1: OUT_PULSE<br>//DIO10 / DIO11: timer cross zero output pulse.<br>//(out_width effective)<br>// 2: OUT_LEVEL<br>//DIO10 / DIO11: timer cross zero output will //make.<br>u8 Tout_mode,<br>// 0: NO_TOUT ,<br>// DIO10 / DIO11: use as general digital output<br>// 1: OUT_PULSE<br>//DIO10 / DIO11: timer cross zero output pulse.<br>//(out_width effective)<br>// 2: OUT_LEVEL<br>//DIO10 / DIO11: timer cross zero output will //make.<br>// 4:OUT_TOGGLE<br>//DIO10 / DIO11: timer cross zero toggles output<br><br>u16 Tout_width,<br>// Output pulse width based on 1us clock, only<br>//valid in Tout_mode is OUT_PULSE<br><br>u8 cont_single,<br>// 0: SINGLE_CYCLE |

| | | |
|---|---|---|
| | | //single cycle mode, timer will stop operation<br>//when time constant count down to zero.<br>// 1: ALWAYS_RUN<br>//continuous operation mode, timer will reload<br>//time constant and continue operation when<br>//time constant count down to zero.<br>} |

- **EX-94421A_counter_set**

**Format : u32 status = EX94421A_counter_set (u8 CardID, u8 TimerID,**
**Counter_struct *TC_struct)**

**Purpose:** To setup counter operation mode or update counter

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| TC_struct | struct | struct TC_struct<br>{<br>u8 TiGate_MODE,<br>// 0: NO_GATE<br>//Always count without gate function,<br>//Timer/counter0, Timer/counter1:<br>//DIO00, DIO01 is counter pulse input<br>//DIO02, DIO03 is digital input.<br><br>// 1:GATED<br>//DIO00, DIO01 is counter pulse input (Ti)<br>//DIO02, DIO03 is gate input, internal logic active //high will pass the counter Ti pulse to counter //after command start_TC<br><br>// 2: EDGE_START<br>//DIO00, DIO01 is counter pulse input (Ti)<br>//DIO02, DIO03 is gate input, internal logic active //high will start topass the counter Ti pulse to //counter after command start_TC<br><br>u32 counter_constant,<br>// Counter constant<br><br>u8 Tout_mode,<br>// 0: NO_TOUT<br>// Timer/counter0, Timer/counter1:<br>// DIO10, DIO11 use as general digital output<br><br>// 1: OUT_PULSE<br>//DIO10, DIO11: timer cross zero output pulse. |

| | | |
|---|---|---|
| | | //(out_width effective)<br><br>// 2: OUT_LEVEL<br>//DIO10, DIO11: timer cross zero output will //make.<br>// 4:OUT_TOGGLE<br>//DIO10, DIO11: timer cross zero toggles output<br><br>u16 Tout_width,<br>// Output pulse width based on 1us clock, only<br>//valid in Tout_mode is OUT_PULSE<br><br>u8 cont_single<br>// 0: SINGLE_CYCLE<br>//single cycle mode, counter will stop operation //when time constant count down to zero.<br>// 1: ALWAYS_RUN<br>// continuous operation mode, counter will reload //time constant and continue operation when time //constant count down to zero.<br>} |

## EX-94421A_PWM_set

**Format :** **u32 status = EX94421A_PWM_set(u8 CardID, u8 TimerID, PWM_struct *PWM_struct)**

**Purpose:** To setup PWM operation mode or update PWM.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| PWM_struct | struct | PWM_struct<br>{<br>u8 TiGate_MODE,<br>// 0: NO_GATE<br>//Always count without gate function,<br>//Timer/counter0, Timer/counter1:<br>//DIO00, DIO01 is counter pulse input<br>//DIO02, DIO03 is digital input.<br><br>// 1:GATED<br>//DIO00, DIO01 is counter pulse input (Ti)<br>//DIO02, DIO03 is gate input, internal logic active<br>//high will pass the counter Ti pulse to counter<br>//after command start_TC<br><br>u16 PWM_freq;<br>// PWM frequency clock count based on 33MHz<br>// clock<br><br>u16 PWM_duty;<br>//PWM duty clock count based on 33MHz clock<br><br>//DIO10, DIO11 use as PWM output<br>} |

**Note:**

1. PWM base clock is based on33MHz, say if you want your PWM frequency is 20KHz, please put the PWM_freq = (33MHz/20KHz) = 1650
2. PWM duty must less than PWM_freq for proper operation, from the example above, the PWM_duty value can be 1 ~ 1649. For 50% duty, the PWM_duty will be 1650/2 = 825

- **EX-94421A_quadrature_set**

**Format :** **u32 status = EX94421A_quadrature_set (u8 CardID,u8 TimerID,**
**u8 Multiple_rate)**

**Purpose:** To setup quadrature counter operation mode

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| Multiple_rate | u8 | Only valid for quadrature mode, in other mode, this parameter is ignored.<br>0: MULTIPLE_4 (default)<br>A,B phase input multiple rate is 4<br>1: MULTIPLE_2<br>A,B phase input multiple rate is 2<br>2: MULTIPLE_1<br>A,B phase input multiple rate is 1 |

**Note:**

1. Port0 is forced to be input.
2. DIO00 is A phase input and DIO02 is B phase input for counter0.
3. DIO01 is A phase input and DIO03 is B phase input for counter1.

- **EX-94421A_TC_start**

**Format :** **u32 status = EX94421A_TC_start (u8 CardID,u8 TimerID)**

**Purpose:** To start timer/counter/PWM/quadrature counter operation mode

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |

- **EX-94421A_TC_stop**

**Format :** **u32 status = EX94421A_TC_stop (u8 CardID, u8 TimerID)**

**Purpose:** To stop timer/counter/PWM/quadrature counter operation mode

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY switch |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |

- **EX-94421A_TC_set**

**Format :** **u32 status=EX94421A_TC_set (u8 CardID, u8 TimerID, u8 index,u32 data)**

**Purpose:** To set data to counter/timer register

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| index | u8 | 0: TC_CONTROL<br>1: TC_MODE<br>2: TiGate_MODE<br>3: To_MODE<br>4: RETRIGGER_MODE<br>5: PRELOAD<br>6: COUNTER<br>7: OUT_WIDTH<br>8: MULTIPLE_RATE |
| data | u32 | register data to be set |

**Note:**

1. please refer the next segment “Note: Meaning of setting or return value of different index”
2. Write to IRQ_STATUS will reset the corresponding bit. Say, if write with bit0=1, the status bit0 will reset but other bit will not effect.

- **EX-94421A_TC_read**

**Format :** **u32 status=EX94421A_TC_read (u8 CardID, u8 TimerID, u8 index,u32 *data)**

**Purpose:** To read data from counter/timer

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | 0: timer/counter0<br>1: timer/counter1 |
| index | u8 | 0: TC_CONTROL<br>1: TC_MODE<br>2: TiGate_MODE<br>3: Tout_MODE<br>4: RETRIGGER_MODE<br>5: PRELOAD<br>6: COUNTER<br>7: OUT_WIDTH<br>8: MULTIPLE_RATE |

**Output:**

| Name | Type | Description |
|---|---|---|
| data | u32 | Data read back |

**Note:**

Meaning of setting or return value of different index

| index | register | value | meaning |
|---|---|---|---|
| 0 | TC_CONTROL | 0 | STOP, stop operation of TC |
| | | 1 | START, start operation of TC |
| 1 | TC_MODE | 0 | TIMER_MODE |
| | | 1 | COUNTER_MODE |
| | | 3 | SW_DEC (a write will software decrease counter by 1 and return to COUNTER_MODE.) |
| | | 4 | PWM_MODE |
| | | 8 | QUADRATURE_MODE |
| 2 | TiGate_MODE | 0 | NO_GATE |
| | | 1 | GATED |
| | | 2 | EDGE_START |
| 3 | Tout_MODE | 0 | NO_TOUT |
| | | 1 | OUT_PULSE |
| | | 2 | OUT_LEVEL |
| | | 4 | OUT_TOGGLE |
| 4 | RETRIGGER_MODE | 0 | SINGLE_CYCLE |
| | | 1 | ALWAYS_RUN |
| 5 | PRELOAD | 1~0xffffffff | Counter or timer or PWM preload value |
| 6 | COUNTER | 1~0xffffffff | Set (write): will write preload and counter<br>Read : will read counter on the fly |
| 7 | OUT_WIDTH | 1~0xffff | Output pulse width based on 1us |
| 8 | MULTIPLE_RATE | 0 | 0: MULTIPLE_4 (default)<br>A,B phase input multiple rate is 4 |
| | | 1 | 1: MULTIPLE_2<br>A,B phase input multiple rate is 2 |
| | | 2 | 2: MULTIPLE_1<br>A,B phase input multiple rate is 1 |

- **EX-94421A_TC_input_polarity_set**

**Format :** **u32 status = EX94421A_TC_input_polarity_set (u8 CardID,u8 TimerID, u8 input,u8 polarity)**

**Purpose:** Set TC input polarity.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | timer/counter designation<br>0: timer/counter0<br>1: timer/counter1 |
| input | u8 | input:<br>0: Gate.<br>1: Ti (clock input). |
| polarity | u8 | polarity values:<br>0: means normal.<br>1: means invert. |

**Note:**

timer/counter0, DIO00 clock input , DIO02 gate input.

timer/counter1, DIO01 clock input , DIO03 gate input.

- **EX-94421A_TC_input_polarity_read**

**Format :** **u32 status = EX94421A_TC_input_polarity_read (u8 CardID,u8 TimerID, u8 input,u8 *polarity)**

**Purpose:** Read TC input polarity.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | timer/counter designation<br>0: timer/counter0<br>1: timer/counter1 |
| input | u8 | input:<br>0: Gate.<br>1: Ti (clock input). |

**Output:**

| Name | Type | Description |
|---|---|---|
| polarity | u8 | polarity values:<br>0: means normal.<br>1: means invert. |

- **EX-94421A_TC_output_polarity_set**

**Format :** **u32 status = EX94421A_TC_output_polarity_set (u8 CardID,u8 TimerID, u8 polarity)**

**Purpose:** Set TC output polarity

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | timer/counter designation<br>0: timer/counter0<br>1: timer/counter1 |
| polarity | u8 | TC output polarity:<br>0: means normal<br>1: means invert |

**Note:**

timer/counter0, DIO10 Signal output.

timer/counter1, DIO11 Signal output.

- **EX-94421A_TC_output_polarity_read**

**Format :** **u32 status = EX94421A_TC_output_polarity_read (u8 CardID,u8 TimerID, u8 * polarity)**

**Purpose:** Read TC output polarity

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| TimerID | u8 | timer/counter designation<br>0: timer/counter0<br>1: timer/counter1 |

**Output:**

| Name | Type | Description |
|---|---|---|
| polarity | u8 | TC output polarity:<br>0: means normal<br>1: means invert |

## 5.5 Interrupt function

Interrupt is a efficient method to quick response without occupy too much system resource. EX-94421A provide timer/counter and TTL port0 as interrupt source, to use interrupt function use

***EX94421A_IRQ_link_process( )*** to link your irq service routine,

***EX94421A_IRQ_enable( )*** to enable it and

***EX94421A_IRQ_disable( )*** to disable it.

***EX94421A_IRQ_mask_set( )*** to mask off the undesired source;

***EX94421A_IRQ_mask_read( )*** to read back for verify the mask setting

***EX94421A_IRQ_IO_polarity_set( )*** to set the polarity of port0 IRQ generation.

***EX94421A_IRQ_IO_polarity_read( )*** to read back for verifying.

After you enable and link interrupt, you can enable/disable timer/counter function or enable/disable interrupt function as you need.

To check the irq status

***EX94421A_IRQ_status_read( )*** will do.

- **EX-94421A_IRQ_link_process**

**Format :** **u32 status = EX94421A_IRQ_link_process(u8 CardID,**
**void (__stdcall *callbackAddr(u8 CardID))**

**Purpose:** To link the interrupt source with the callback function.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| callbackAddr | void | the address of your callback function |

- **EX-94421A_IRQ_enable**

**Format :** **u32 status = EX94421A_IRQ_enable (u8 CardID,HANDLE *phEvent)**

**Purpose:** Enable interrupt.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |

**Output:**

| Name | Type | Description |
|---|---|---|
| phEvent | HANDLE | The returned handle of event |

- **EX-94421A_IRQ_disable**

**Format :** **u32 status = EX94421A_IRQ_disable (u8 CardID)**

**Purpose:** To disable interrupt.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |

- **EX-94421A_IRQ_mask_set**

**Format :** **u32 status = EX94421A_IRQ_mask_set(u8 CardID,u16 mask)**

**Purpose:** Set IRQ mask.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |
| mask | u16 | b0:<br>0, disable DIO00 as interrupt source<br>1, enable DIO00 as interrupt source<br>…<br>b6:<br>0, disable DIO06 as interrupt source<br>1, enable DIO06 as interrupt source<br>b7:<br>0, disable DIO07 as interrupt source<br>1, enable DIO07 as interrupt source<br>b8:<br>0, disable TC0 counter counts to zero as interrupt source<br>1, enable TC0 counter counts to zero as interrupt source<br>b9:<br>0, disable TC1 counter counts to zero as interrupt source<br>1, enable TC1 counter counts to zero as interrupt source |

- **EX-94421A_IRQ_mask_read**

**Format :** **u32 status = EX94421A_IRQ_mask_read(u8 CardID,u16 *mask)**

**Purpose:** Read IRQ mask.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |

**Output:**

| Name | Type | Description |
|---|---|---|
| mask | u16 | b0:<br>0, disable DIO00 as interrupt source<br>1, enable DIO00 as interrupt source<br>…<br>b6:<br>0, disable DIO06 as interrupt source<br>1, enable DIO06 as interrupt source<br>b7:<br>0, disable DIO07 as interrupt source<br>1, enable DIO07 as interrupt source<br>b8:<br>0, disable TC0 counter counts to zero as interrupt source<br>1, enable TC0 counter counts to zero as interrupt source<br>b9:<br>0, disable TC1 counter counts to zero as interrupt source<br>1, enable TC1 counter counts to zero as interrupt source |

- ## EX-94421A_IRQ_IO_polarity_set

**Format :** **u32 status =EX94421A_IRQ_IO_polarity_set (u8 CardID, u8 polarity)**

**Purpose:** Sets the interrupt polarity of TTL port0

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |
| polarity | u8 | port0 polarity values:<br>0: any bit of port0 form low to high (default) can generate IRQ<br>1: any bit of port0 from high to low can generate IRQ |

- ## EX-94421A_IRQ_IO_polarity_read

**Format :** **u32 status = EX94421A_IRQ_IO_polarity_read (u8 CardID, u8 * polarity)**

**Purpose:** Read the I/O IRQ polarity of the TTL port0.

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by Rotary SW |

**Output:**

| Name | Type | Description |
|---|---|---|
| polarity | u8 | port0 polarity values:<br>0: any bit of port0 form low to high(default) can generate IRQ<br>1: any bit of port0 from high to low can generate IRQ |

## EX-94421A_IRQ_status_read

**Format :** **u32 status = EX94421A_IRQ_status_read(u8 CardID,u16 * state)**

**Purpose:** To read IRQ state

**Parameters:**

**Input:**

| Name | Type | Description |
|---|---|---|
| CardID | u8 | assigned by DIP/ROTARY SW |

**Output:**

| Name | Type | Description |
|---|---|---|
| state | u16 | Bit 0: DIO00 generate IRQ.<br>…<br>Bit 7: DIO07 generate IRQ.<br>Bit 8: timer/counter0 generate IRQ.<br>Bit 9: timer/counter1 generate IRQ. |

**Note:**

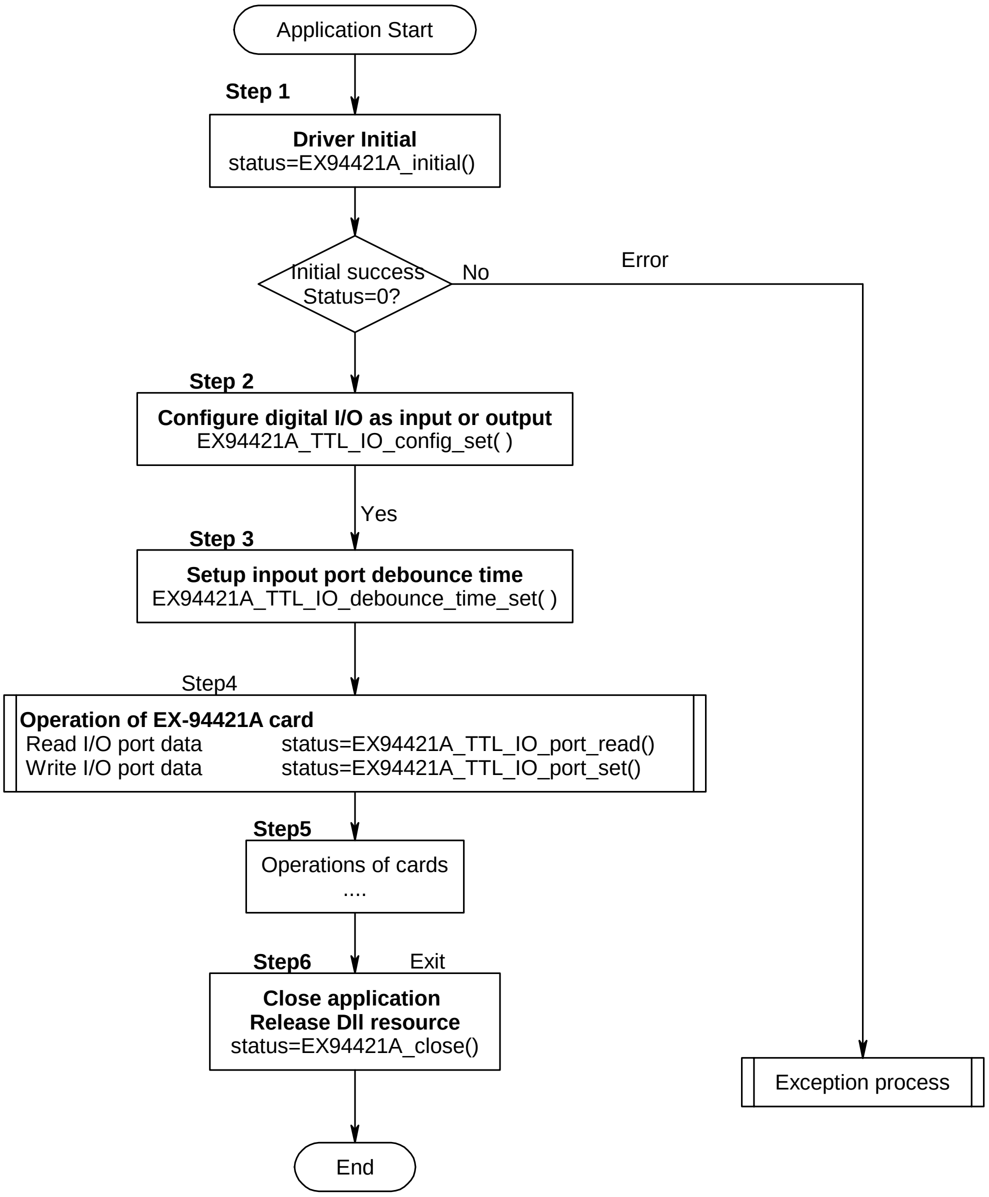
This command will also clear the on board IRQ_status register, the second read will not be correct.

## 5.6 Error conditions

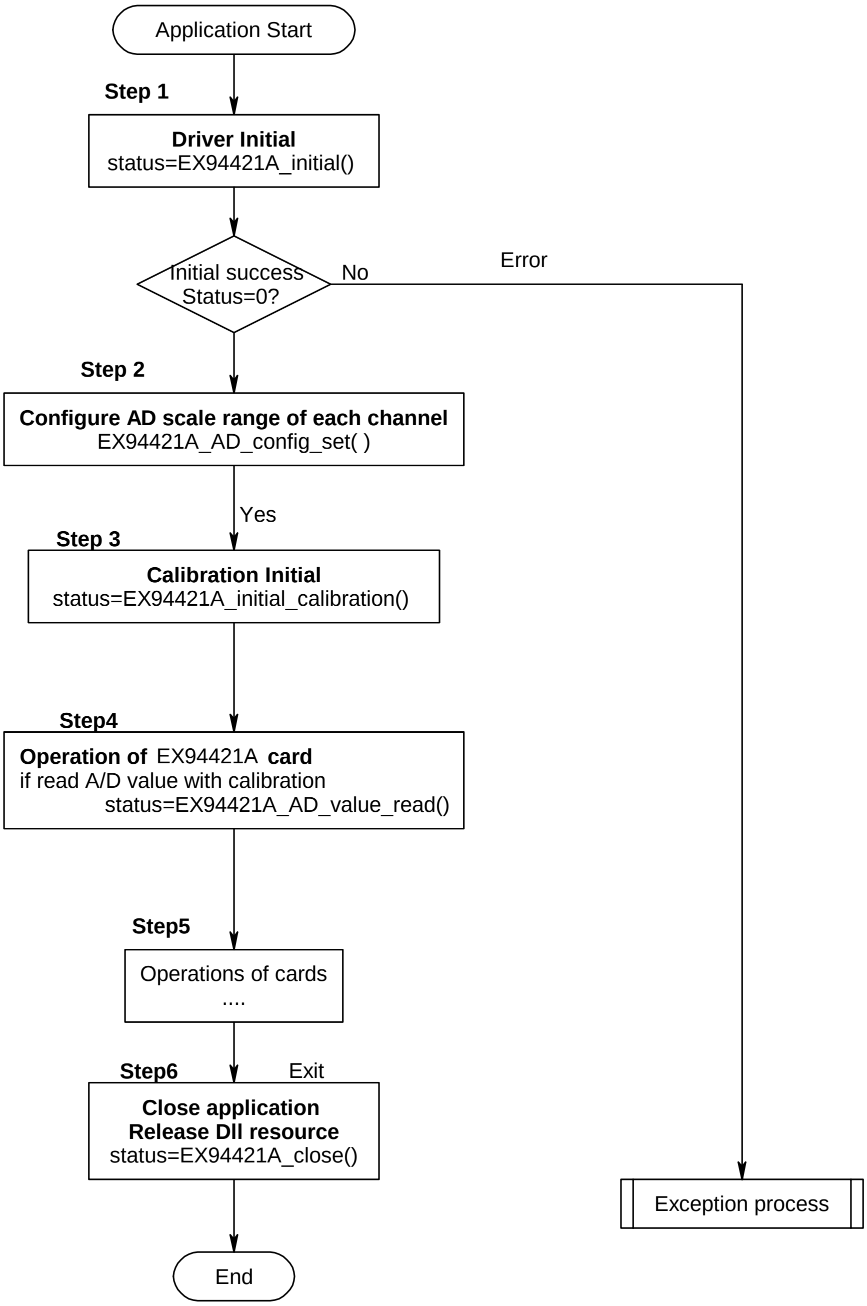
These error types may indicate an internal hardware problem on the board. Error Codes summary contains a detailed listing of the error status returned by EX-94421A functions.

# 6. Flow chart of application implementation

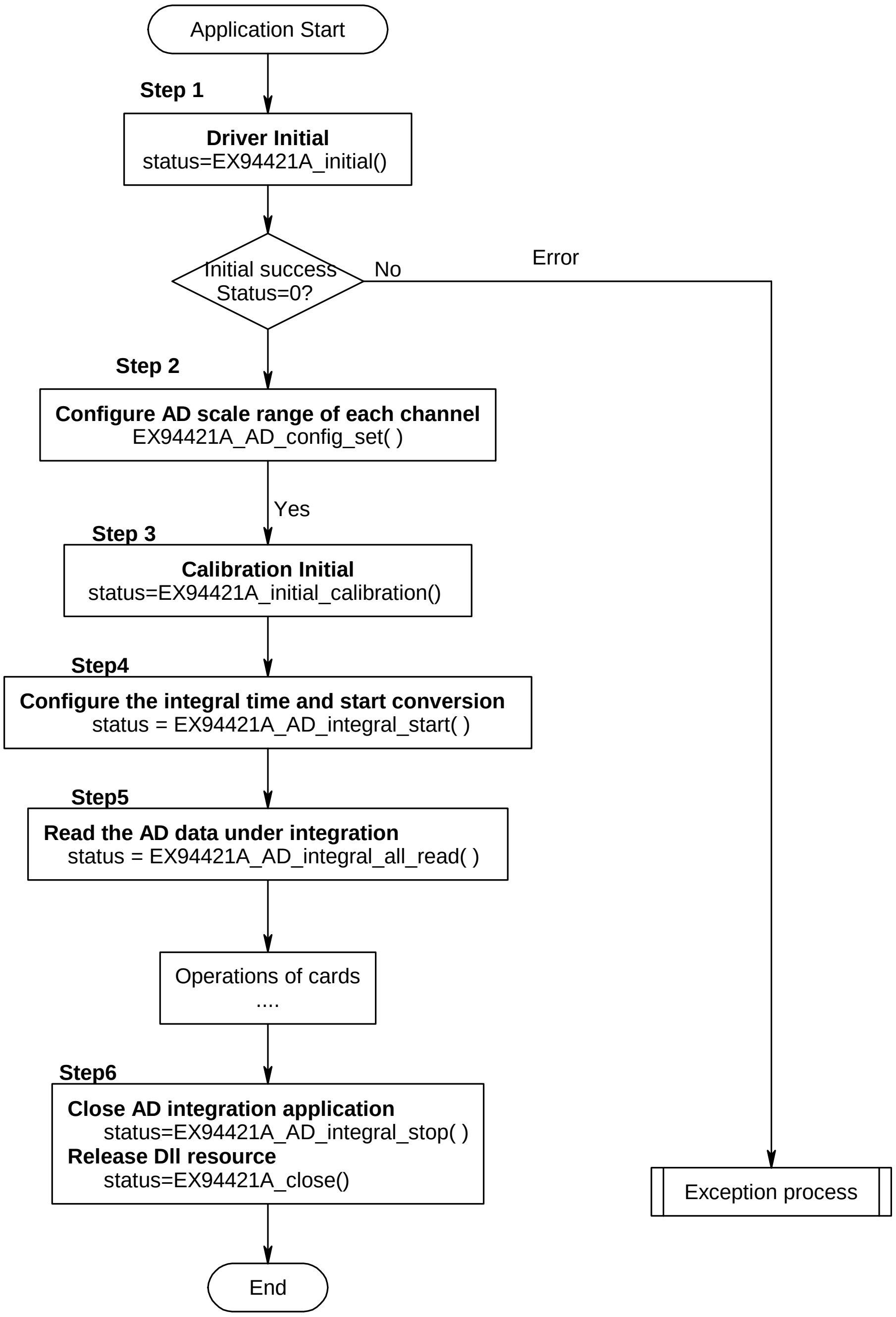
## 6.1 EX-94421A Flow chart of digital I/O application implementation


Application Start
Step 1
Driver Initial
status=EX94421A_initial()
Initial success
Status=0?
No
Error
Step 2
Configure digital I/O as input or output
EX94421A_TTL_IO_config_set( )
Yes
Step 3
Setup inpout port debounce time
EX94421A_TTL_IO_debounce_time_set( )
Step4
Operation of EX-94421A card
Read I/O port data
status=EX94421A_TTL_IO_port_read()
Write I/O port data
status=EX94421A_TTL_IO_port_set()
Step5
Operations of cards
....
Step6
Exit
Close application
Release Dll resource
status=EX94421A_close()
End
Exception process

## 6.2 EX-94421A Flow chart of analog I/O application implementation

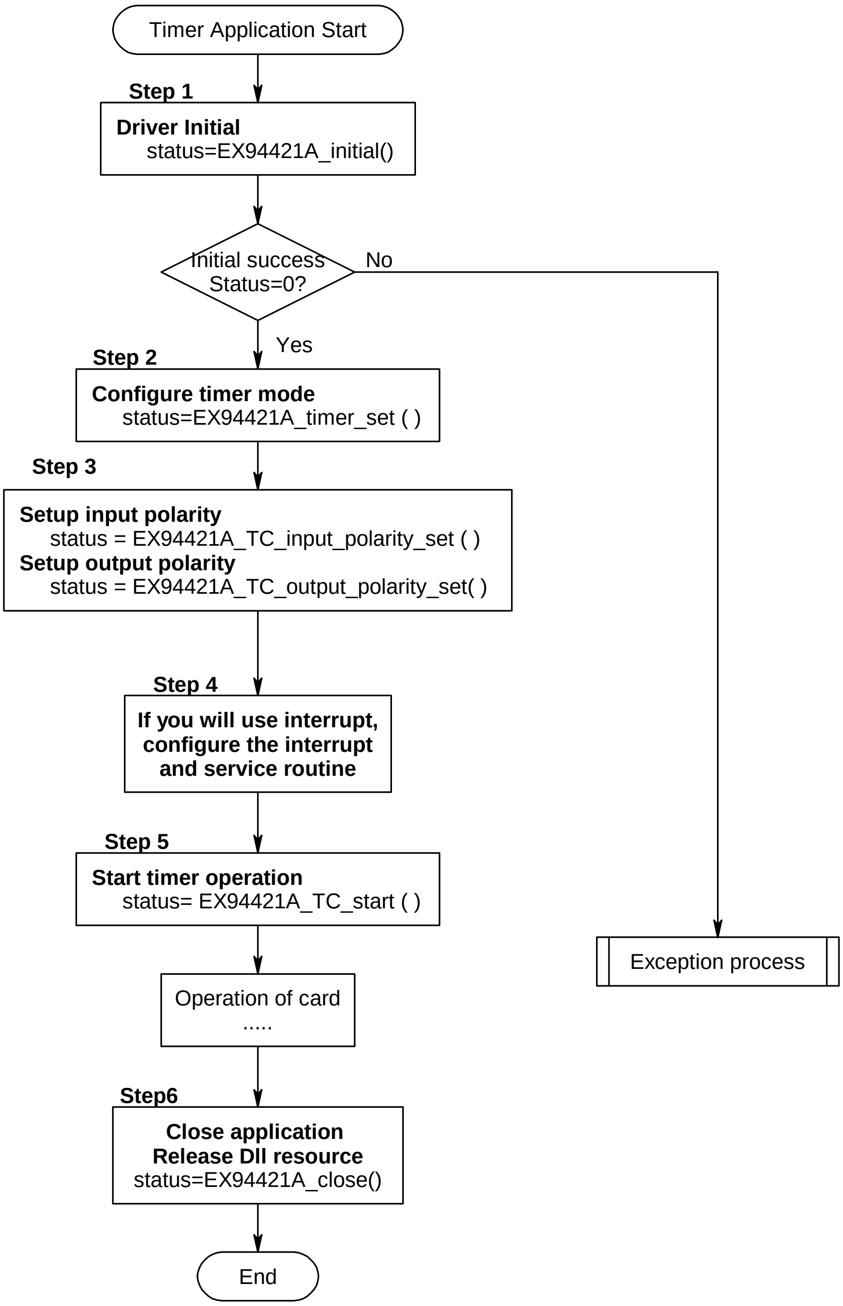


## 6.3 EX-94421A Flow chart of analog I/O application with embedded integration function

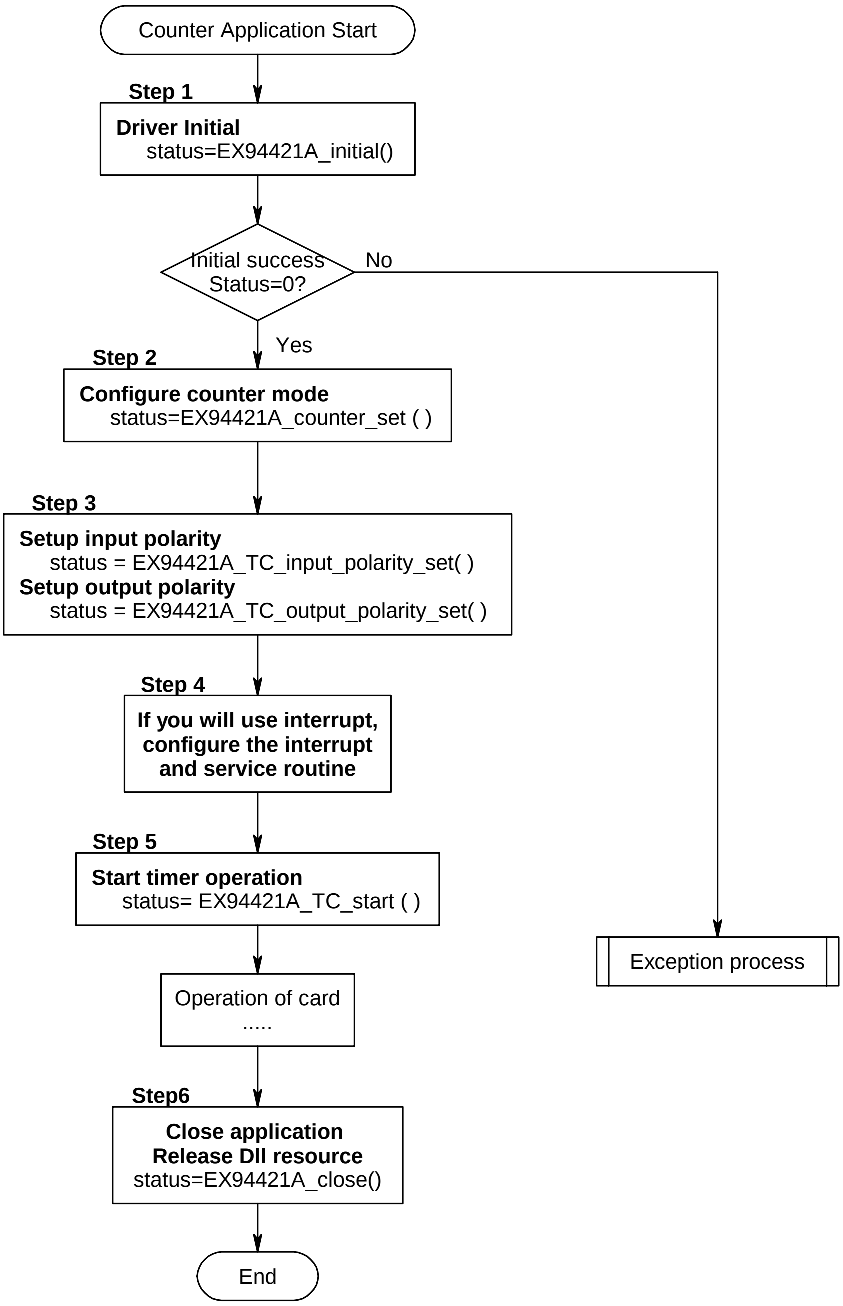


Application Start
Step 1
Driver Initial
status=EX94421A_initial()
Initial success
Status=0?
No
Error
Step 2
Configure AD scale range of each channel
EX94421A_AD_config_set( )
Yes
Step 3
Calibration Initial
status=EX94421A_initial_calibration()
Step4
Configure the integral time and start conversion
status = EX94421A_AD_integral_start( )
Step5
Read the AD data under integration
status = EX94421A_AD_integral_all_read( )
Operations of cards
....
Step6
Close AD integration application
status=EX94421A_AD_integral_stop( )
Release Dll resource
status=EX94421A_close()
Exception process
End

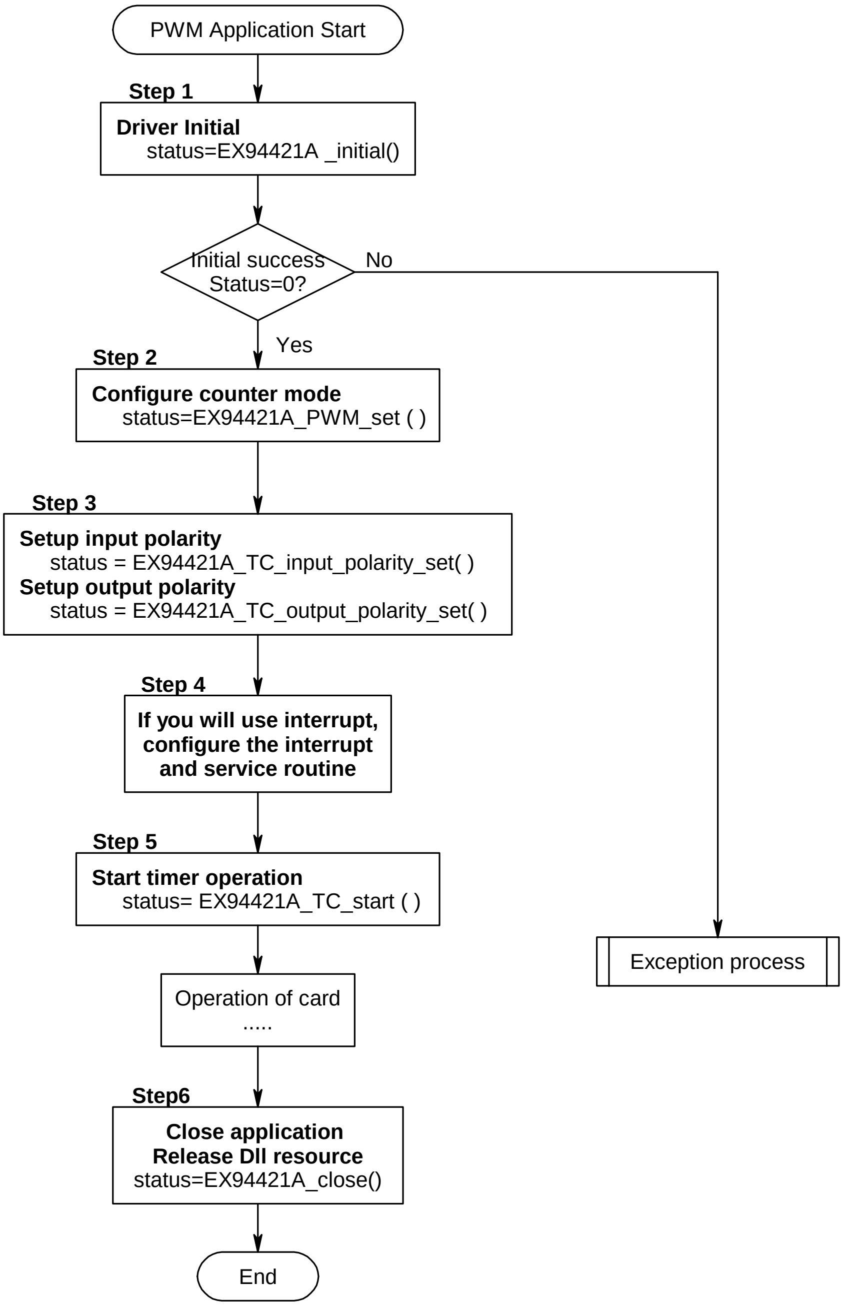
## 6.1 EX-94421A Flow chart of Timer application



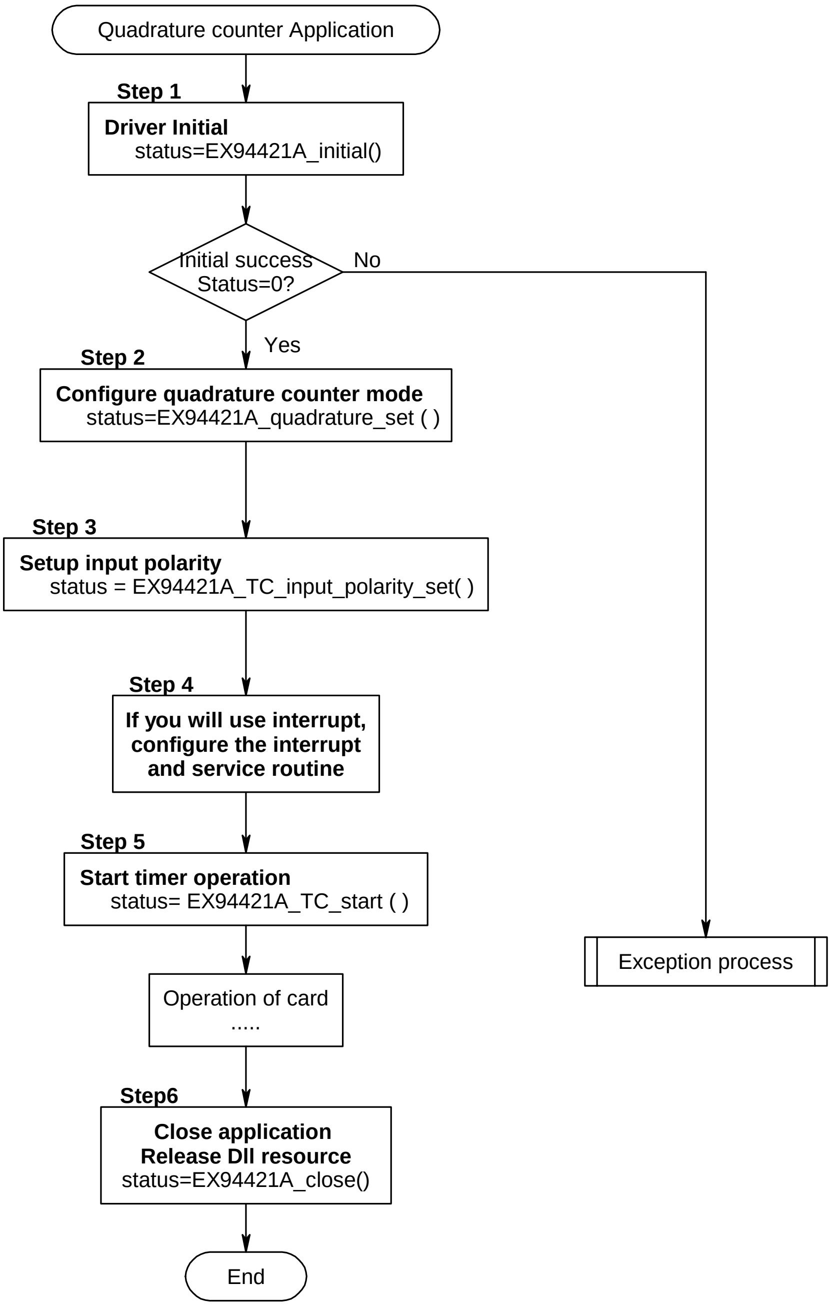
## 6.2 EX-94421A Flow chart of Counter application



## 6.3 EX-94421A Flow chart of PWM application



## 6.4 EX-94421A Flow chart of quadrature counter application


Quadrature counter Application
Step 1
Driver Initial
status=EX94421A_initial()
Initial success
Status=0?
No
Yes
Step 2
Configure quadrature counter mode
status=EX94421A_quadrature_set ( )
Step 3
Setup input polarity
status = EX94421A_TC_input_polarity_set( )
Step 4
If you will use interrupt, configure the interrupt and service routine
Step 5
Start timer operation
status= EX94421A_TC_start ( )
Exception process
Operation of card
.....
Step6
Close application
Release Dll resource
status=EX94421A_close()
End

## 7. Dll list

| | Function Name | Description |
|---|---|---|
| 1. | EX94421A_initial( ) | Card initial. |
| 2. | EX94421A_close( ) | Card Close. |
| 3. | EX94421A_info( ) | Read Card Address. |
| | | |
| 4. | EX94421A_AD_config_set( ) | Set AD config. |
| 5. | EX94421A_AD_config_read( ) | Read AD config. |
| 6. | EX94421A_AD_value_read( ) | Read AD value. |
| 7. | EX94421A_AD_data_read( ) | Read AD data value. |
| 8. | EX94421A_AD_integral_start( ) | start AD conversion with integral constant |
| 9. | EX94421A_AD_integral_all_read( ) | read port integral result of AD conversion data |
| 10. | EX94421A_AD_integral_stop( ) | stop AD integral conversion |
| | | |
| 11. | EX94421A_TTL_IO_config_set( ) | Set port config (input or output) |
| 12. | EX94421A_TTL_IO_config_read( ) | Read port config (input or output) |
| 13. | EX94421A_TTL_IO_port_set( ) | Write data to port |
| 14. | EX94421A_TTL_IO_port_read( ) | Read port data |
| 15. | EX94421A_TTL_IO_point_set( ) | Write bit of data to port |
| 16. | EX94421A_TTL_IO_point_read( ) | Read data of a specific point |
| 17. | EX94421A_TTL_IO_debounce_time_set( ) | Write Ti point debounce time. |
| 18. | EX94421A_TTL_IO_debounce_time_read( ) | Read debounce time. |
| | | |
| 19. | EX94421A_timer_set( ) | setup timer operation mode or update timer |
| 20. | EX94421A_counter_set( ) | setup counter operation modeor update counter |
| 21. | EX94421A_PWM_set( ) | setup PWM operation mode or update PWM |
| 22. | EX94421A_quadrature_set( ) | setup quadrature counter operation mode |
| 23. | EX94421A_TC_start( ) | start timer/counter/PWM/quadrature counter operation mode |
| 24. | EX94421A_TC_stop( ) | stop timer/counter/PWM/quadrature counter operation mode |
| 25. | EX94421A_TC_set( ) | set data to counter/timer register |
| 26. | EX94421A_TC_read( ) | read data from counter/timer register |
| 27. | EX94421A_TC_input_polarity_set( ) | Set TC input polarity |
| 28. | EX94421A_TC_input_polarity_read( ) | Read back TC input polarity setting |
| 29. | EX94421A_TC_output_polarity_set( ) | Set TC output polarity |
| 30. | EX94421A_TC_output_polarity_read( ) | Read back TC output polarity setting |
| | | |
| 31. | EX94421A_IRQ_link_process( ) | Link process IRQ. |
| 32. | EX94421A_IRQ_enable( ) | Enable IRQ. |
| 33. | EX94421A_IRQ_disable( ) | Disable IRQ. |
| 34. | EX94421A_IRQ_mask_set( ) | Set mask. |
| 35. | EX94421A_IRQ_mask_read( ) | Read mask. |
| 36. | EX94421A_IRQ_IO_polarity_set( ) | Set IO polarity. |
| 37. | EX94421A_IRQ_IO_polarity_read( ) | Read IO polarity. |
| 38. | EX94421A_IRQ_status_read ( ) | Read IRQ status. |
| | | |

# 8. EX-94421A Error codes summary

## 8.1 EX-94421A Error codes table

| Error Code | Symbolic Name | Description |
|---|---|---|
| 0 | DRV_NO_ERROR | Success, No error. |
| 2 | DRV_INIT_ERROR | Driver initial error |
| 3 | DRV_UNLOCK_ERROR | Security unclock failure |
| 4 | DRV_LOCK_COUNTER_ERROR | Dead lock, unclock failure more than 10 times |
| 5 | SDRV_SET_SECURITY_ERROR | Password overwrite error |
| 100 | DEVICE_RW_ERROR | Device Read/Write error |
| 101 | DRV_NO_CARD | No EX-94421A card on the system. |
| 102 | DRV_DUPLICATE_ID | EX-94421A CardID duplicate error. |
| 104 | DRV_PAR_ERROR | Bad parameter or illegal parameter |
| 300 | EXDIO_ID_ERROR | Function input parameter error. CardID setting error, CardID doesn’t match the DIP/ROTARY SW setting |
| 301 | EXAIO_MODE_ERROR | Mode parameter error.<br>Parameter out of range. |
| 302 | EXAIO_CHANNEL_ERROR | Channel parameter error.<br>Parameter out of range. |
| 305 | EXAIO_CONVERSION_ERROR | Conversion time over. Maybe no hardware or bad hardware. |
| 306 | EXAIO_CONVERSION_BUSY | A/D is busy in conversion |
| 400 | EXAIO_PORT_ERROR | Port parameter error.<br>Parameter out of range. |
| 401 | EXAIO_STATE_ERROR | State parameter error.<br>Parameter out of range. |
| 402 | EXAIO_POINT_ERROR | Point parameter error.<br>Parameter out of range. |
| 403 | EXAIO_EEPROM_RW_ERROR | Eeprom R/W error |
| 405 | EXAIO_CALIBRATION_ERROR | Calibration error.<br>Maybe out of range. |
| 406 | EXAIO_TIMERID_ERROR | TimerID parameter error.<br>Parameter out of range. |
| 407 | EXAIO_TO_MODE_ERROR | To_mode parameter error.<br>Parameter out of range. |
| 408 | EXAIO_TI_MODE_ERROR | Ti_mode parameter error.<br>Parameter out of range. |
| 409 | EXAIO_PARAMETER_ERROR | Parameter error.<br>Parameter out of range. |
| 500 | EXAIO_EVENT_ERROR | Event error.<br>Maybe no event created. |