User's Manual

32

S3CTRL Driver for Linux

User's Manual: Software

R-Car H2/M2 Series

Renesas Electronics
www.renesas.com

Rev.1.00 Mar. 2014

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

How to Use This Manual

Purpose and Target Readers

This manual is designed to provide the user with an understanding the functions of S3CTRL Driver for Linux. This manual is written for engineers who use S3CTRL Driver.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Please refer to documents of software and hardware for a target system implementing S3CTRL Driver as necessary.
The following documents are related documents. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	**Description**	**Document Title**	**Notes**
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	R-Car {H/M}2 User's Manual: Hardware	
User's manual for Software	Description of Memory Manager	Memory Manager for Linux	
	Description of S3CTRL Driver	S3CTRL Driver for Linux	This manual
	Description of R-Car H2/M2 Linux BSP	Linux Interface Specification Start-Up Guide	

Notation of Numbers and Symbols

This manual use following notation.

Binary	0bXXXXXXXX	(X=0 or 1)
Decimal	XXX	(X=0-9)
Hex	0xXXXXXXXX	(X=0-9,A-F)

List of Abbreviations and Acronyms

Abbreviation	Full Form
MMNGR	Memory Manager
S3CTL	S3CTRL Driver
DTV	Digital Terrestrial Television
DVD	Digital Versatile Discs

Table Of Contents

S3CTRL Driver for Linux
User's Manual: Software

May. 2014

1. Overview

1.1. Overview of this Software

・S3CTRL Driver (hereinafter called in S3CTL) is the function that provides the TL conversion.

1.2. Configuration of this Software

This software consists of the following resources.

- ・Document
- ・Source code
- ・Sample application
- ・Make file

To use this software, the following additional software which is not included in this software is required.
Details of this additional software are shown below.

- ・Kernel module source code

  This software is distributed based on Dual MIT/GPLv2 licenses.
  Figure 1-2 shows the lists of these source files.

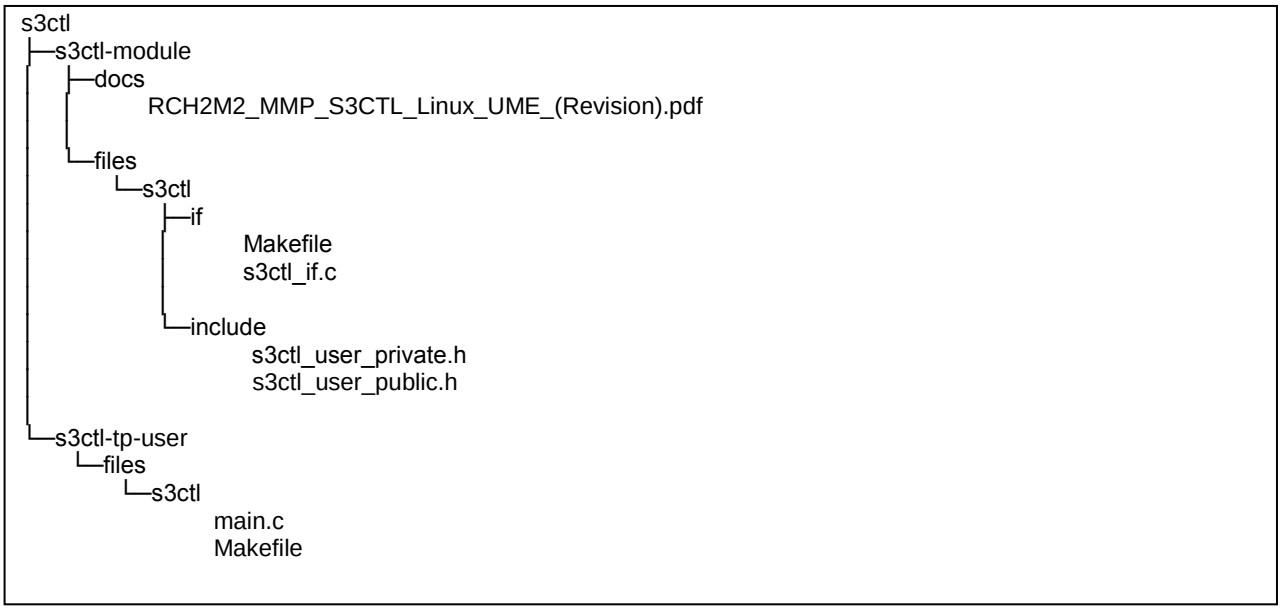


Figure 1-1 Configuration of this software

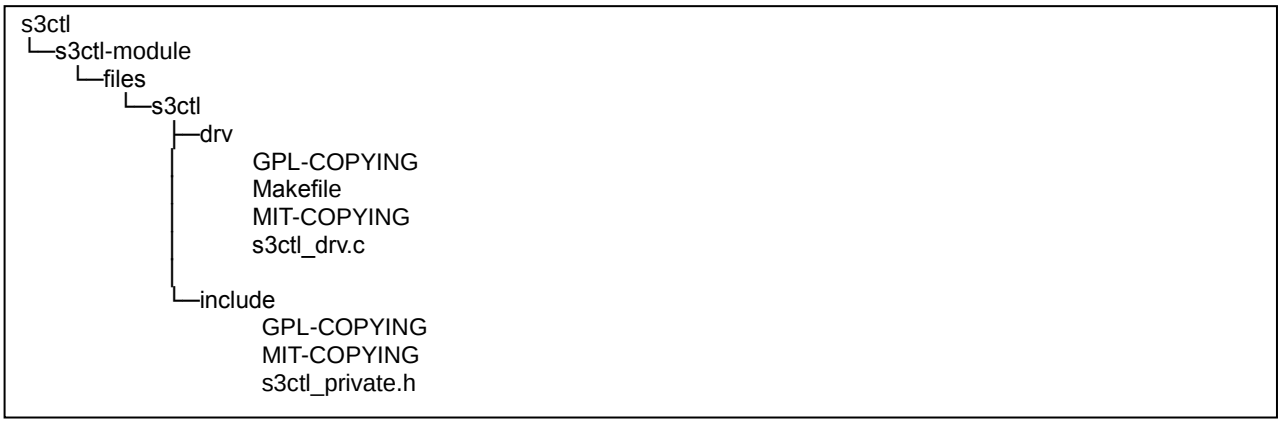


Figure 1-2 Kernel module source code not included in this software

1.3. Development Environments

This section describes the development environments for this software.

1.3.1. Hardware Development Environment

Hardware Name		Remarks
Device	R-Car H2/M2	-
Board	RENESAS LAGER/KOELSH	-

1.3.2. Software Development Environment

Software Name	**Version / Revision**	**Remarks**
R-Car H2/M2 Linux BSP	-	-

2. Installation Procedures

2.1. Building the shared library

The following is the procedure for building the shared library that is included in this software.

1) Setting environment variables

Follow Linux BSP Start-Up Guide.

In addition to the guide, add the path.

```
$export PATH=$WORK/gcc-linaro-arm-linux-gnueabihf-4.7-2013.02-01-20130221_linux/bin:$PATH
$export BUILDDIR=(the include path of the library headers in your environment)
```

2) Building the shared library

Execute make under release/user/if.

```
$cd s3ctl/s3ctl-module/files/s3ctl/if
$make
```

3) Verifying the shared library

Make sure that the following kernel modules are built under release/user/if.

libs3ctl.so.x.x.x ("x.x.x" depends on the version.)

2.2. Building the Kernel Module

The following is the procedure for building the kernel module that is not included in this software.

1) Setting environment variables

Follow Linux BSP Start-Up Guide.

In addition to the guide, set the following value.

```
$export KERNELDIR=(the kernel path in your environment) (for example, $WORK/linux-stable)
$export KERNELSRC=(the directory path storing the include directory of the driver headers
                   in your environment) (for example, $WORK/linux-stable)
$export CP=cp
```

2) Building the kernel modules

Execute make under kernel/driver.

```
$cd s3ctl/s3ctl-module/files/s3ctl/drv
$make
```

3) Verifying the kernel module

Make sure that the following kernel modules are built under kernel/driver.

s3ctl.ko

2.3. Storing the sharead library, the Kernel module

The following procedure shows storing the shared librarie and the kernel module that are built at 2.1. and 2.2..

1) Storing the shared library

Create /local/lib under /usr in the rootfs of the target board.

```
$mkdir /usr/local
$mkdir /usr/local/lib
```

Store libs3ctl.so.x.x.x under /usr/local/lib in the rootfs of the target board.

```
$cp libs3ctl.so.x.x.x /usr/local/lib
```

2) Storing the kernel module

Store the kernel module under the directory in the rootfs of the target board.

```
$mkdir /tmp (/tmp is a example. Threfore change the directory accoding to your environment.)
$cp s3ctl.ko /tmp
```

3) Setting environment variable

Set /usr/local/lib to LD_LIBRARY_PATH.

```
$export LD_LIBRARY_PATH=/usr/local/lib
```

4) Copying the link to the target rootfs.

Copy the links to the rootfs of the target board at PC.

```
$cp -d libs3ctl.so.x (the rootfs of the target board)/usr/local/lib/
$cp -d libs3ctl.so (the rootfs of the target board)/usr/local/lib
```

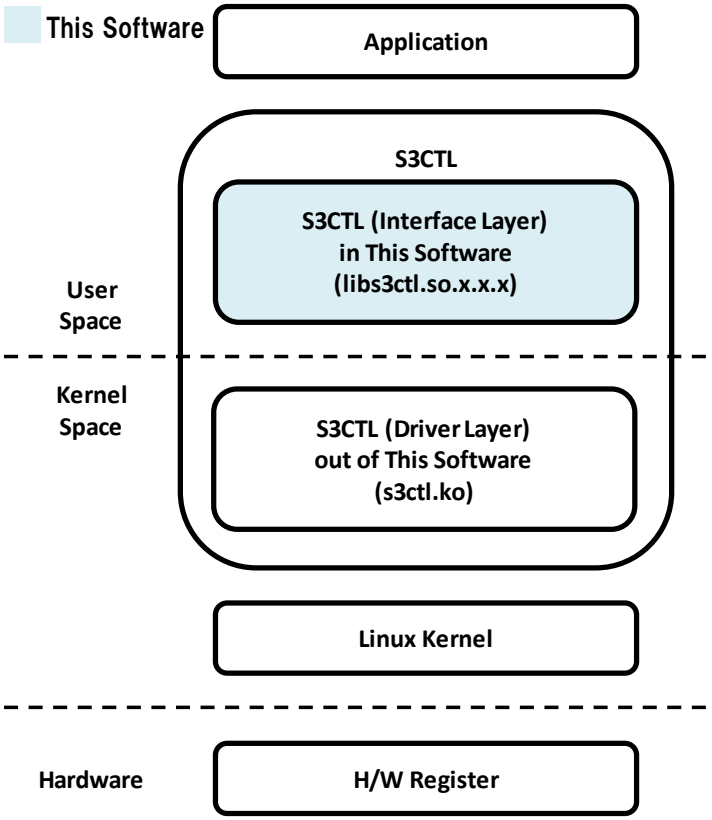
5) Install the kernel module at the target board.

Execute insmod under the directory storing the kernel module.

```
$cd /tmp
$insmod s3ctl.ko
```

3. Module Configuration

The module configuration of S3CTL is as follows.



・S3CTL consists of the interface layer and the driver layer.
Interface Layer (in this software)
This layer is in the user space and provides API.
Driver Layer (out of this software)
This layer is in the kernel space and sets the hardware registers.

・The execution context of S3CTL is the process or the thread calling S3CTL.
That is, S3CTL does not have the process and the thread.

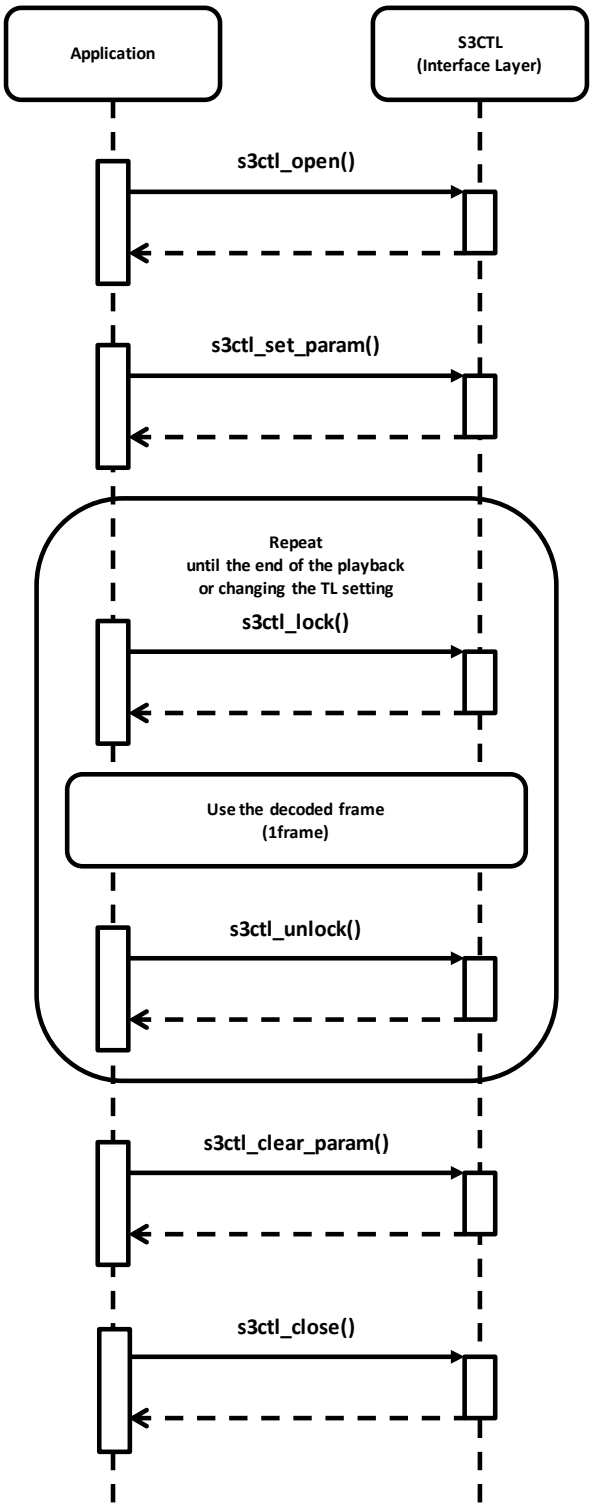
4. List of API

No	Name	Function
1	s3ctl_open()	Open S3CTL
2	s3ctl_close()	Close S3CTL
3	s3ctl_set_param()	Set the parameter of the TL conversion
4	s3ctl_clear_param()	Clear the parameter of the TL conversion
5	s3ctl_lock()	Lock S3CTL
6	s3ctl_unlock()	Unlock S3CTL
7	s3ctl_get_param()	Get the parameter of the TL conversion

5. Sample Sequence

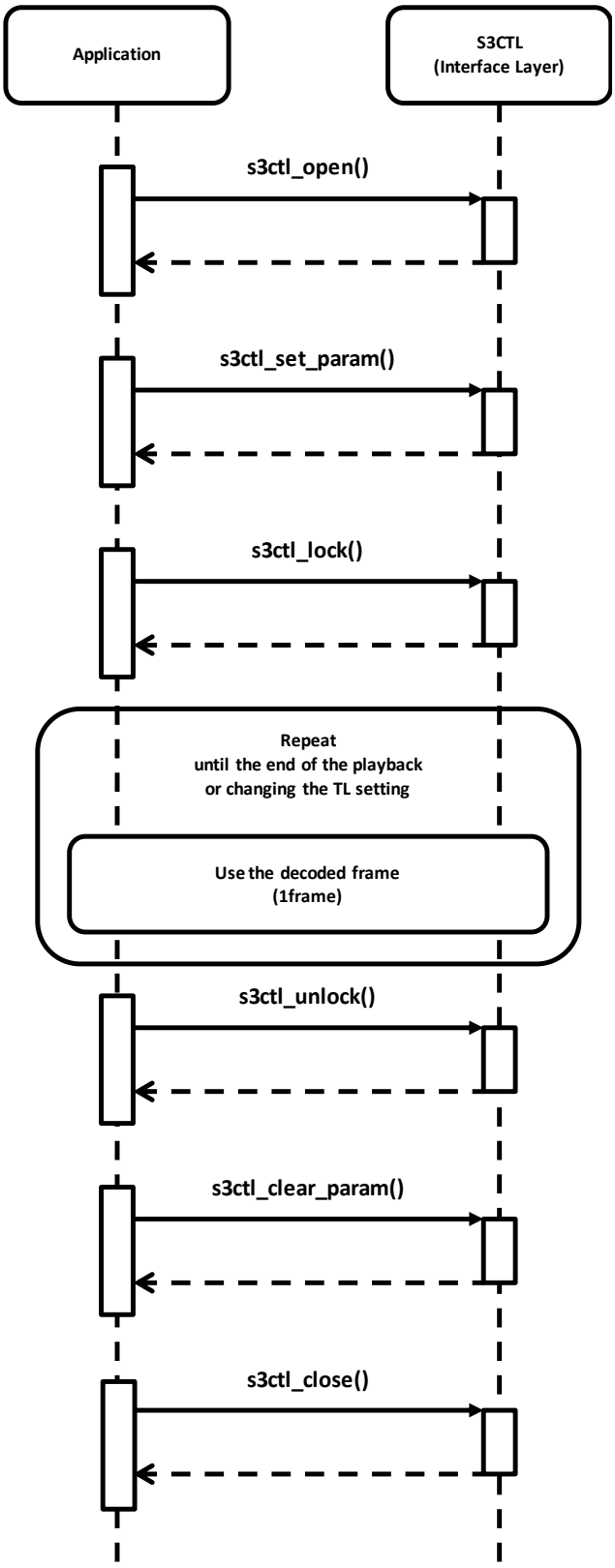
Case of OMX

Call s3ctl_set_param() before the input of the decoded frame to H/W IP.
After s3ctl_set_param(), per one frame, call s3ctl_lock() before the input of the decoded frame.
Call s3ctl_unlock() after the completion of H/W IP.



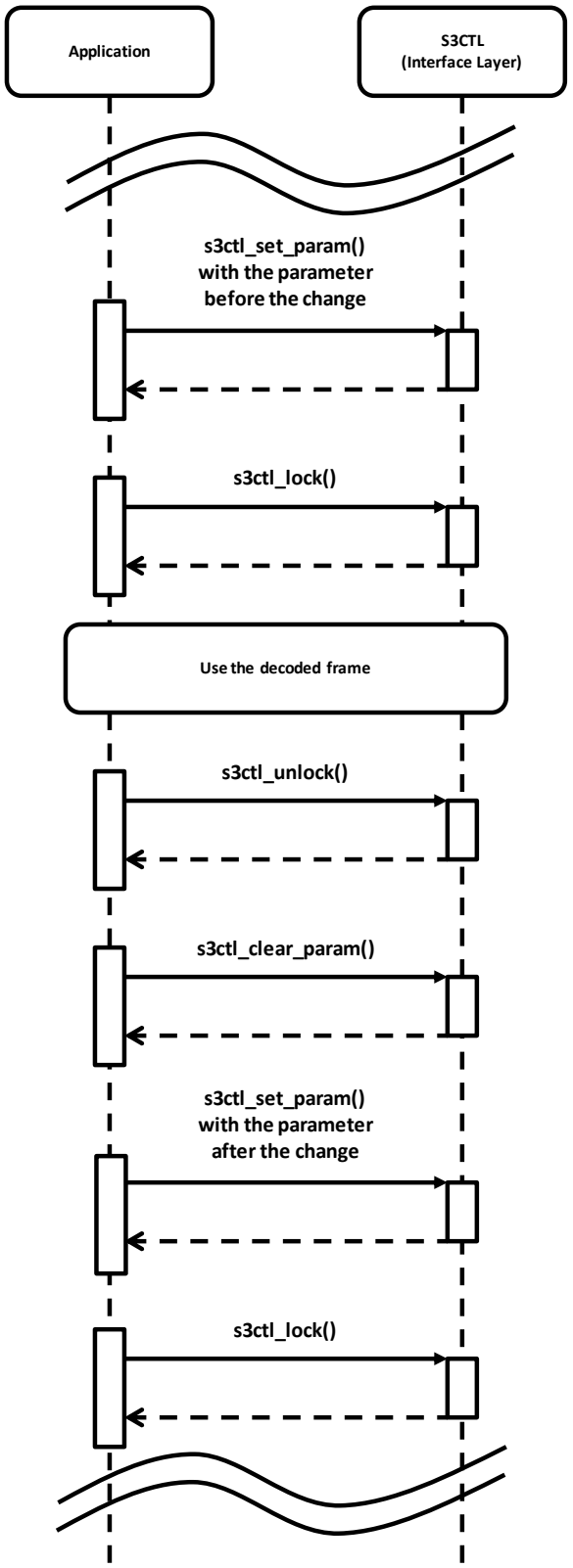
Case of DTV or DVD

Call the sequence until s3ctl_lock() before the input of the decoded frame to H/W IP.
Call the sequence from s3ctl_unlock() after stopping the input to H/W IP.



Case of Chage the parameter of the TL conversion

Change the parameters with s3ctl_set_param() in the OPEN state after calling s3ctl_unlock() and s3ctl_clear_param()



6. State Chart and State Matrix

The state is managed in each ID allocated by s3ctl_open().
The number of the states is 4.

INIT

INIT is the state before S3CTL is opened.
INIT is the state of convenience.
Threfore S3CTL does not have this state.

OPEN

OPEN is the state after S3CTL is opened and ID is allocated,
the state before the parameter of the TL conversion is set.

SET

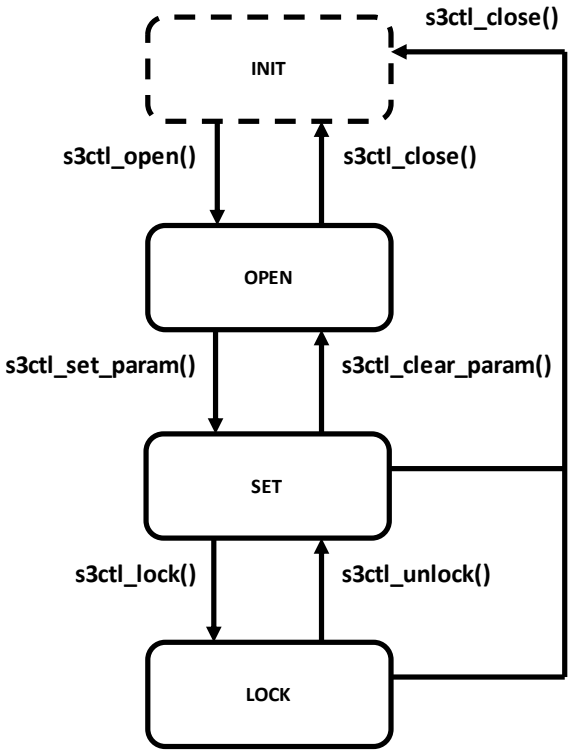
SET is the state after the parameter of the TL conversion is set.

LOCK

LOCK is the state after the parameter of the TL conversion is set to the register.
That is, LOCK is the state when the conversion is executed.

The reason why there are SET and LOCK is as follows.

The parameter of the TL conversion is set to the register, when the frame is read.
However the parameter is changed, when the stream information is changed.
Threfore the reason is why SET and LOCK enable the application to use the register
in a time-division without setting the same parameter in each frame.



○:Sequence OK ×:Sequence NG -:Irrealizable Sequence

State / API	INIT	OPEN	SET	LOCK
s3ctl_open()	○ ⇒OPEN	-	-	-
s3ctl_close()	×	○ ⇒INIT	○ ⇒INIT	○ ⇒INIT
s3ctl_set_param()	×	○ ⇒SET	×	×
s3ctl_clear_param()	×	×	○ ⇒OPEN	×
s3ctl_lock()	×	×	○ ⇒LOCK	×
s3ctl_unlock()	×	×	×	○ ⇒SET
s3ctl_get_param()	×	○	○	○

7. API Specification

7.1. Open S3CTL

Name

s3ctl_open

Synopsis

```
int s3ctl_open(
    S3CTL_ID *pid,          (output)
)
```

Arguments

S3CTL_ID *pid,	The address storing ID of the TL conversion

Struct

-

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_PARE	Parameter Error (Condition) When the application set NULL to *pid*.

Description

(1) This function allocates ID of the TL conversion.

(2) ID allocated by this function is held until s3ctl_close() is called.

Notes

(1) Allocate the space to set to pid in the application.

The type of the space set to *pid* is S3CTL_ID.

Example

```
S3CTL_ID id;
s3ctl_open(&id);
```

(2) Don't call this API at the context of a signal handler.

See Also

-

7.2. Close S3CTL

Name

s3ctl_close

Synopsis

```
int s3ctl_close(
    S3CTL_ID id                    (input)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion

Struct

-

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.

Description

(1) This function frees *id*.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) Don't call this API at the context of a signal handler.

7.3. Set the Parameter of the TL Conversion

Name

s3ctl_set_param

Synopsis

```
int s3ctl_set_param(
    S3CTL_ID id,                    (input)
    struct S3_PARAM *param          (input)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion
struct S3_PARAM **param*	The parameter of the TL conversion

Struct

struct S3_PARAM {

unsigned long	*phy_addr*;	The physical address of the TL conversion that the lower 12bit address is shifted to right
unsigned long	*stride*;	The stride of the TL space S3_STRIDE_128 S3_STRIDE_256 S3_STRIDE_512 S3_STRIDE_1K S3_STRIDE_2K S3_STRIDE_4K
unsigned long	*area*;	The size of the TL space [KB] S3_AREA_256 S3_AREA_512 S3_AREA_1K S3_AREA_2K S3_AREA_4K S3_AREA_8K S3_AREA_16K S3_AREA_32K S3_AREA_64K S3_AREA_128K S3_AREA_256K

};

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_PARE	Parameter Error (Condition) (1) When the application sets NULL to *param*. (2) When the application sets 0 to *phy_addr* (3) When the upper 9bits of *phy_addr* are not 0. (4) When *phy_addr* * 4096 is not aligned by *stride* * 32. (5) When the application set the undefined value in *stride* of (3) Struct to *stride*. (6) When the application set

	the undefined value in *area* of (3) Struct to *area*.
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.

Description

(1) This function set the parameter of the TL conversion corresponding to *id*.

(2) The parameter set by this function is not set to the register until s3ctl_lock() is called.
That is, the TL conversion is not executed until s3ctl_lock() is called.

(3) The parameter set by this function is held until s3ctl_clear_param() or s3ctl_close() is called.

(4) S3CTL sets 128 x 32 to the tile type internally.

(5) The unit of *area* is KB.
Threfore when the application sets S3_AREA_256K to *area*,
The size of the TL space is 256K [KB].
That is, the size is 256MB.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) Set the space not to overlap the space set by the other ID to *phy_addr, area*.

(3) Align the physical address of the space allocated by MMNGR, and set the aligned address to *phy_addr*.
Allocate "the definition of *area* in (3) Struct" + "the alignment adjustment Size" from MMNGR.
For example, when the size of the TL space is 800KB, the size set to MMNGR is not 800KB + the align adjustment size,
but 1MB (= S3_AREA_1K[KB]) + the align adjustment size.
Align the top of the address by *stride* * 32.
For example, when the alignment adjustment size is 32KB (= stride 1KB * 32) and the size of the TL space is 800KB,
Step.1 Allocate 1024KB + (32KB -1) from MMNGR.
Step.2 Align the physical address allocated at Step.1 by 32KB. *1
Step.3 Set the address aligned at Step.2 to *phy_addr*. *2

(4) Allocate the space of *param* in the application.

*1 The address of the physical space allocated by MMNGR at Step.1 is not the physical address but the address that MMNGR shifts the physical address to 12bits right.
Threfore when you align the address, shift the address of the physical space

allocated by MMNGR to 12bits left, and align the shifted address.

*2 *phy_addr* is not the physical address but the address that MMNGR shifts the physical address to 12bits right.
Threfore shift the aligned address at Step.2 to 12bits right, and set the address to *phy_addr*.

(5) Don't call this API at the context of a signal handler.

See Also

-

7.4. Clear the Parameter of the TL Conversion

Name

s3ctl_clear_param

Synopsis

```
int s3ctl_clear_param(
      S3CTL_ID id                    (input)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion

Struct

-

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.

Description

(1) This function clear the TL parameter of *id* to 0.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) Don't call this API at the context of a signal handler.

See Also

-

7.5. Lock S3CTL

Name

s3ctl_lock

Synopsis

```
int s3ctl_lock (
    S3CTL_ID id                    (input)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion

Struct

-

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.
R_S3_BUSY	Busy (Condition) When the register of the TL conversion is used.

Description

(1) This function set the parameter of the TL conversion corresponding to *id* to the register.
That is, this function execute the TL conversion in based on the parameter corresponding to *id*.

(2) The value set to the register is valid until s3ctl_unlock() is called.

(3) The number of the register is 8.
Threfore after the states of 8 different ids are all LOCK, when the application calls this function, this function returns R_S3_BUSY.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) After the return value is R_S3_BUSY, if you wants to continue the playback, call this function again.

(3) Don't call this API at the context of a signal handler.

See Also

-

7.6. Unlock S3CTL

Name

s3ctl_unlock

Synopsis

```
int s3ctl_unlock (
    S3CTL_ID id                    (input)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion

Struct

-

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.

Description

(1) This function stops the TL conversion.

(2) This function only stops the TL conversion by setting to the register.
That is, this function does not clear the parameter of the TL conversion corresponding to *id*.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) Don't call this API at the context of a signal handler.

See Also

-

7.7. Get the Parameter of the TL Conversion

Name

s3ctl_get_param

Synopsis

```
int s3ctl_get_param (
    S3CTL_ID id,                    (input)
    struct S3_PARAM *param          (output)
)
```

Arguments

S3CTL_ID *id*	ID of the TL conversion
struct S3_PARAM **param*	The address storing the parameter of the TL conversion

Struct

Refer to 7.4

Return Value

R_S3_OK	Normal End
R_S3_FATAL	Fatal Error
R_S3_SEQE	Sequence Error (Condition) When this function is called in the state where × is written in the state chart.

Description

(1) This function get the parameter of the TL conversion corresponding to *id*.
The parameter got by this function is the parameter set
by s3ctl_set_param() or the parameter cleared by s3ctl_clear_param().
That is, the parameter got by this function is not always the parameter set
to the register.

Notes

(1) Set ID, which is allocated by s3ctl_open() and is not freed, to *id*.

(2) Allocate the space of *param* in the application.

(3) Don't call this API at the context of a signal handler.

See Also

-

8. Header Files

Include s3ctl_public.h in the application, when the application calls the APIs of S3CTL.

9. Definition

9.1. Return Value Definition

Definition	Value	Content
R_S3_OK	0	Normal End
R_S3_FATAL	-1	Fatal Error
R_S3_SEQE	-2	Sequence Error
R_S3_PARE	-3	Parameter Error
R_S3_BUSY	-4	Busy

9.2. Parameter Definition

Definition	Value	Content
S3_STRIDE_128	128	The value of the stride
S3_STRIDE_256	256	
S3_STRIDE_512	512	
S3_STRIDE_1K	1024	
S3_STRIDE_2K	2048	
S3_STRIDE_4K	4096	
S3_AREA_256	0	The size of the TL space
S3_AREA_512	1	
S3_AREA_1K	2	
S3_AREA_2K	3	
S3_AREA_4K	4	
S3_AREA_8K	5	
S3_AREA_16K	6	
S3_AREA_32K	7	
S3_AREA_64K	8	
S3_AREA_128K	9	
S3_AREA_256K	10	

Revision History	S3CTRL Driver for Linux User's Manual: Software

Rev.	Date	Description	
		Page	Summary
1.00	May.2014	—	First edition issued

S3CTRL Driver for Linux User's Manual: Software

Publication Date: Rev.1.00 May. 2014

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

S3CTRL Driver for Linux
User's Manual: Software

Renesas Electronics Corporation