

CONFIDENTIAL

RH850/D1x Device Family
Renesas Graphics Library
HyperBus Controller (HYPB) Driver
User's Manual: Software

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

CONFIDENTIAL

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
 3. 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 shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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; industrial robots; etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 9. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
 11. This document shall not be reprinted, 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.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

CONFIDENTIAL

Trademark

- Green Hills, the Green Hills logo, INTEGRITY, MULTI, DoubleCheck, EventAnalyzer, Integrate, SuperTrace, ResourceAnalyzer, CodeFactor, INTEGRITY MULTIVISOR, GMART, GSTART, G-Cover, PathAnalyzer, GHNet, TimeMachine, μ -veLoSiTy, Padded Cell, TotalDeveloper, and Optimizing Compiler are trademarks or registered trademarks of Green Hills Software in the US and/or internationally.
- This software contains the technology owned by TES Electronic Solutions GmbH. All rights reserved for TES Electronic Solutions GmbH
- Trademarks and trademark symbols (® or ™) are omitted in the text of this manual.

CONFIDENTIAL

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding the functions of HYPB driver. This manual is written for engineers who use HYPB 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 drivers and hardware for a target system implementing HYPB as necessary.

The following documents are related documents. Make sure to refer to the latest versions of these documents.

Document Type	Description	Document Title	Document No.
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description	RH850/D1L/D1M Group User's Manual: Hardware	R01UH0451EJ0220
User's manual for Software	Description of RGL overview	Renesas Graphics Library User's Manual: Software	R01US0181ED0400
	Description of WM	Renesas Graphics Library Window Manager (WM) Driver User's Manual: Software	LLWEB-10035990
	Description of SPEA	Renesas Graphics Library Sprite Engine A (SPEA) Driver User's Manual: Software	LLWEB-10035991
	Description of VDCE	Renesas Graphics Library Video Data Controller E (VDCE) Driver User's Manual: Software	LLWEB-10035992
	Description of VOWE	Renesas Graphics Library Video Output Warping Engine (VOWE) Driver User's Manual: Software	LLWEB-10035993
	Description of JCUA	Renesas Graphics Library JPEG Codec Unit A (JCUA) Driver User's Manual: Software	LLWEB-10035994
	Description of SFMA	Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver User's Manual: Software	LLWEB-10064753
	Description of HYPB	Renesas Graphics Library HyperBus Controller (HYPB) Driver User's Manual: Software	LLWEB-10064754 (This manual)
	Description of OCTA	Renesas Graphics Library OctaBus Controller (OCTA) Driver User's Manual: Software	LLWEB-10064755
	Description of VOCA	Renesas Graphics Library Video Output Checker A (VOCA) Driver User's Manual: Software	LLWEB-10063801

CONFIDENTIAL

	Description of DISCOM	Renesas Graphics Library Display Output Comparator (DISCOM) Driver User's Manual: Software	LLWEB-10063802
	Description of DRW2D	Renesas Graphics Library 2D Graphics (DRW2D) Driver User's Manual: Software	LLWEB-10059472
Porting Layer Guide	Description of porting layer of RGL	Renesas Graphics Library Porting Layer Guide	LLWEB-10035995

CONFIDENTIAL

2. Notation of Numbers and Symbols

This manual uses the following notation.

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

CONFIDENTIAL

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
API	Application Programming Interface
CA	Command Address
DDR	Double Data Rate.
H/W	Hardware
HYPB	HyperBus
I/O	Input / Output.

All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

1. Overview	3
1.1 Feature and Scope	3
1.2 Component Structure	3
2. Basic Specification	4
2.1 Summary Specification	4
2.2 Reserved Word	4
2.3 Interrupt Handler List	5
2.4 Error Handling	5
2.4.1 Return code	5
2.4.1.1 Parameter level	5
2.4.1.2 Timing level	5
2.4.1.3 System level	5
2.4.1.4 Hardware level	5
2.4.1.5 Device level	5
2.5 State Transition	6
3. Function Description	8
3.1 Fundamental Concepts	8
3.1.1 HYPB unit	8
3.1.2 System Configuration	8
3.1.3 Operating Mode	9
3.1.3.1 External address space mode	9
3.1.3.2 Manual mode	9
3.1.4 Dependence command of the Hyper flash / Hyper RAM	9
3.2 Using the API	10
3.2.1 Initialization / De-Initialization	10
3.2.2 Hyper RAM - External address space mode	10
3.2.3 Hyper Flash - External address space mode	11
3.2.4 Hyper Flash - Manual mode	12
3.3 Device difference	13
3.4 Header File List	13
4. Functions	14
4.1 Function List	14
4.2 HYPB API Functions	15
4.2.1 Basic functions	15
4.2.1.1 R_HYPB_Init	15
4.2.1.2 R_HYPB_DeInit	17
4.2.1.3 R_HYPB_Open	19
4.2.1.4 R_HYPB_Close	21
4.2.1.5 R_HYPB_DataProtect	23
4.2.1.6 R_HYPB_DataErase	25
4.2.1.7 R_HYPB_DataWrite	27
4.2.1.8 R_HYPB_DataRead	29
4.2.1.9 R_HYPB_UserCmdIssue	31
4.2.1.10 R_HYPB_VersionStringGet	33
4.2.1.11 R_HYPB_MacroVersionGet	34
4.2.2 Interrupt functions	35
5. Types	36
5.1 Basic Types	36
5.2 Definition	37

CONFIDENTIAL

Renesas Graphics Library HyperBus Controller (HYPB) Driver

5.3	Enumerated Type	38
5.3.1	r_hypb_Error_t	38
5.3.2	r_hypb_DeviceType_t	39
5.3.3	r_hypb_OperatingMode_t	40
5.3.4	r_hypb_ParamSecPos_t	41
5.3.5	r_hypb_ProtectionMode_t	42
5.3.6	r_hypb_CmdType_t	43
5.4	Structure Type	44
5.4.1	r_hypb_ParamSec_t	44
5.4.2	r_hypb_CmdTransaction_t	46
5.4.3	r_hypb_CmdSequence_t	47
5.4.4	r_hypb_Latency_t	48
5.4.5	r_hypb_Command_t	49
5.4.6	r_hypb_Config_t	51

1. Overview

1.1 Feature and Scope

The HYPB driver is a driver stack, which enables an abstract access to HyperRAM or HyperFlash memory. The abstraction shall simplify the usage by the application developer and also make it possible to use the same API for different hardware.

1.2 Component Structure

The component structure of HYPB is shown in [Figure 1-1](#).

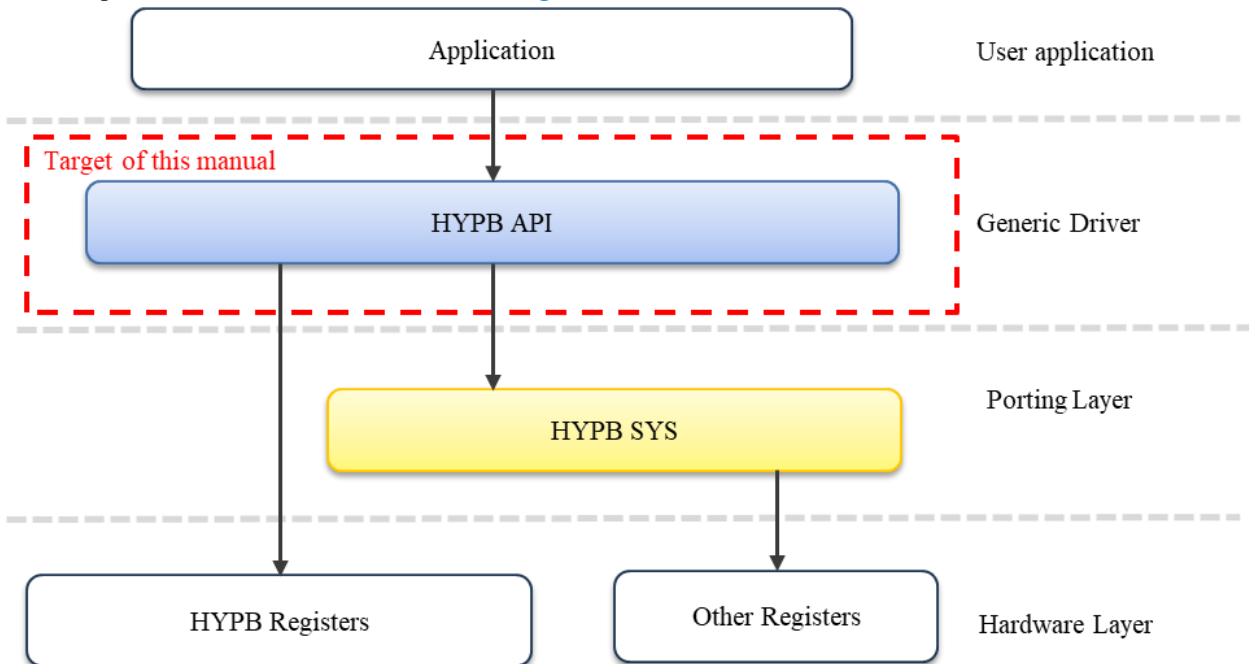


Figure 1-1 Component Structure

For the details of the API, please refer to [Chapter 4](#).

2. Basic Specification

2.1 Summary Specification

The summary of specification is described in [Table 2-1](#).

Table 2-1 Summary Specification

Items	Description
Target LSI	RH850/D1M1-V2, RH850/D1M1A
Main Feature	<ul style="list-style-type: none">• Number of connected devices<ul style="list-style-type: none">○ One Hyper Flash or one Hyper RAM can be connected.• Data bus width<ul style="list-style-type: none">○ 8bits.• Maximum size<ul style="list-style-type: none">○ Maximum flash/RAM size is 16 Mbyte.• Data transfer mode<ul style="list-style-type: none">○ DDR• Operating mode<ul style="list-style-type: none">○ External address space mode○ Manual mode
Semaphore / Mutex	N/A. This can be implemented with porting layer.
Interrupts	N/A.

2.2 Reserved Word

HYPB uses the following prefixes for avoiding confusion from other software. Prefixes of HYPB is described in [Table 2-2](#).

Table 2-2 Prefixes

Prefix	Description
R_HYPB_*	Prefix for HYPB Module
r_hypb_*	

2.3 Interrupt Handler List

None.

2.4 Error Handling

2.4.1 Return code

HYPB driver has 5 types of error codes.

2.4.1.1 Parameter level

Following errors occur by a cause such as abnormality of parameter. In this case, please set valid parameter again.

- R_HYPB_ERR_PARAM_INCORRECT
- R_HYPB_ERR_RANGE_UNIT
- R_HYPB_ERR_RANGE_PARAM

2.4.1.2 Timing level

Following errors occur by a cause such as abnormality of execution timing. In this case, please call again after changing to valid state or timing.

- R_HYPB_ERR_NOT_ACCEPTABLE
- R_HYPB_ERR_COMMAND
- R_HYPB_ERR_LATENCY
- R_HYPB_ERR_PROTECTED
- R_HYPB_ERR_ABORTED
- R_HYPB_ERR_TIMEOUT

2.4.1.3 System level

Following errors occur by a cause such as OS dependent error (e.g. system call error, resource shortage). In this case, please do recovery processing from a system layer, because this status cannot be restored only in this library.

- R_HYPB_ERR_FATAL_OS

2.4.1.4 Hardware level

Following errors occur when unexpected error occurs internally. In this case, please reset the RH850/D1x device.

- R_HYPB_ERR_NG
- R_HYPB_ERR_FATAL_HW

2.4.1.5 Device level

Following errors occur when the function is not supported with target device. In this case, please skip the function call.

- R_HYPB_ERR_DEVICE

2.5 State Transition

Each HYPB unit has following status.

Table 2-3 HYPB unit State Details

No.	State Name	Description
(1)	Uninitialized	Specifies that the HYPB driver is not initialized.
(2)	Initialized	Specifies that the HYPB driver is initialized.
(3)	Idle	Specifies that Manual mode is enabled.
(4)	Executing	Specifies that External address space mode is enabled.

The image describes state transition.

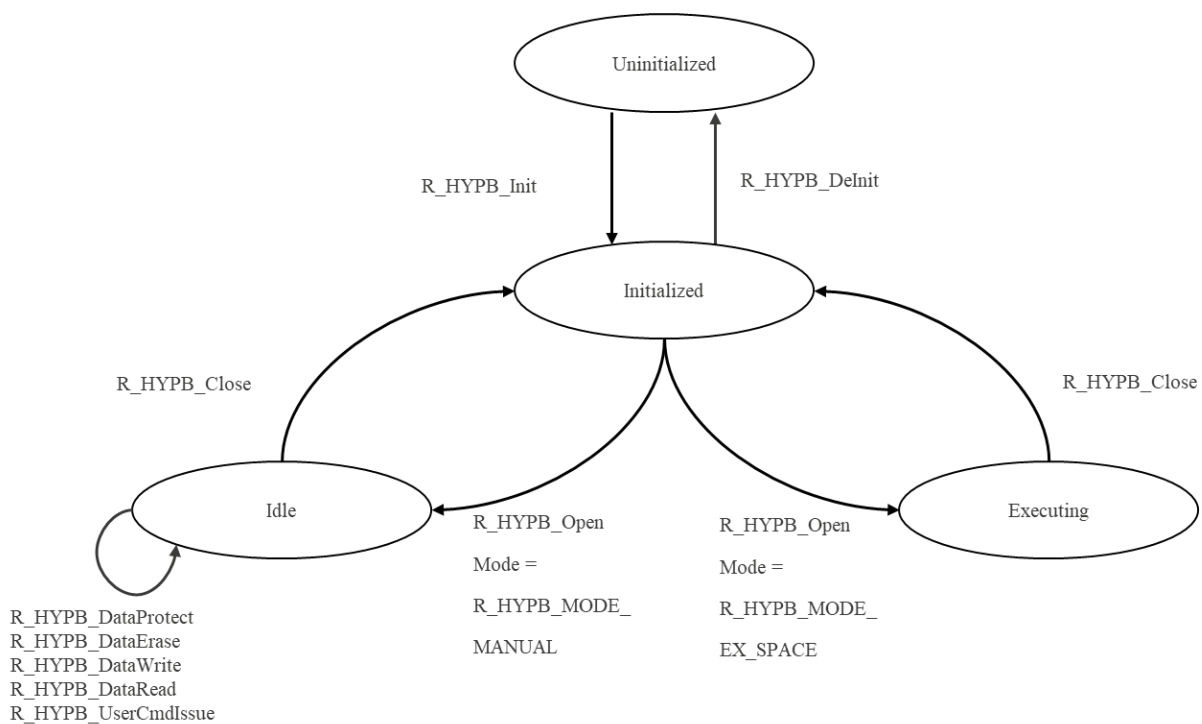


Figure 2-1 State Transition Diagram of HYPB driver

CONFIDENTIAL

Table 2-4 State Transition Table of HYPB unit

Function Name	State			
	Uninitialized	Initialized	Idle	Executing
R_HYPB_Init	OK	NG	NG	NG
R_HYPB_DeInit	NG	OK	NG	NG
R_HYPB_Open	NG	OK	NG	NG
R_HYPB_Close	NG	NG	OK	OK
R_HYPB_DataProtect	NG	NG	OK	NG
R_HYPB_DataErase	NG	NG	OK	NG
R_HYPB_DataWrite	NG	NG	OK	NG
R_HYPB_DataRead	NG	NG	OK	NG
R_HYPB_UserCmdIssue	NG	NG	OK	NG
R_HYPB_VersionStringGet	OK	OK	OK	OK
R_HYPB_MacroVersionGet	OK	OK	OK	OK

3.Function Description

3.1 Fundamental Concepts

3.1.1 HYPB unit

RH850/D1x device has the following number of units of the HYPB.

Table 3-1 Number of units

Feature	RH850/D1x Device Name
	D1M1-V2, D1M1A
HYPB Units	1

Almost HYPB API functions have the argument “Unit”.

User specifies the HYPB H/W unit number to be controlled. The range is only 0.

3.1.2 HYPB Word-address and Word-size

HYPB driver uses Word-address or Word-size. 1 word is 16 bits.

Word-address is 16-bit address. Word-size is 16-bit size. Word-size=1 means 2Byte. Word-size=2 means 4Byte.

3.1.3 System Configuration

This configuration has selected 8 bits data bus width.

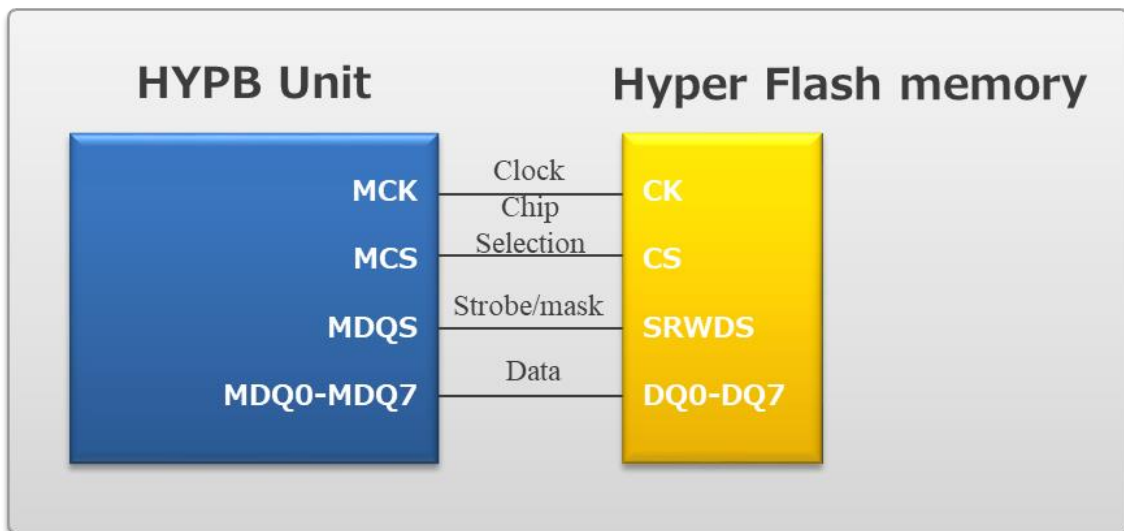


Figure 3-1 System Configuration

3.1.4 Operating Mode

The HYPB driver has two operating modes: external address space read mode and manual mode.

3.1.4.1 External address space mode

A read access from the bus master to the HyperBus memory area automatically converted into a read command to the device and the read data are returned to the bus master. By using this mode, the Hyper RAM/flash memory is assigned to linear memory space and usability is improved.

After R_HYPB_Open is executed with R_HYPB_MODE_EX_SPACE, you can access the memory (16MByte) directly from RH850 CPU. In case of Hyper RAM, Read and Write are possible. In case of Hyper Flash, only Read is possible.

Address space 0x0000_0000 – 0x0100_0000 of Hyper Flash / RAM is mapped to 0x6000_0000 – 0x6100_0000 at address space of RH850/D1x.

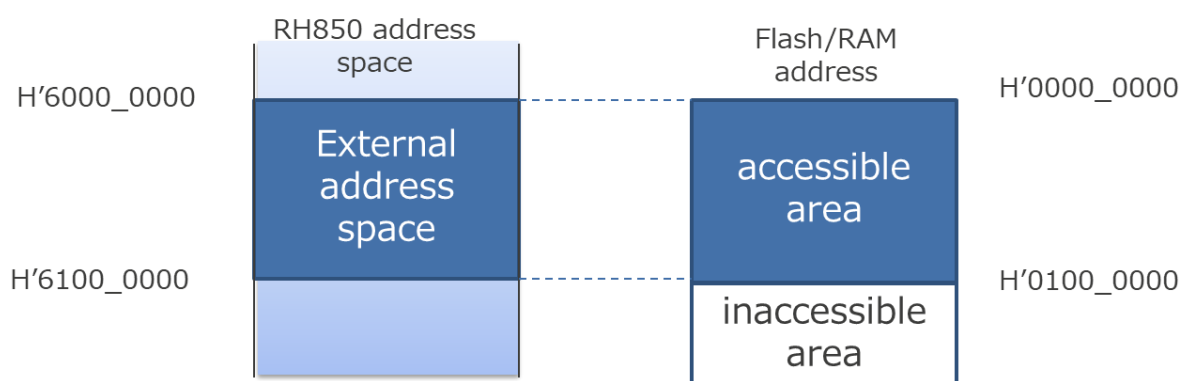


Figure 3-2 Address mapping example

3.1.4.2 Manual mode

Desired command, read, and write accesses to the HyperRAM or HyperFlash memory are possible.

Note: The hyper flash memory has protection function which prohibits writing and erasing. The control method of the protection function is different depending on hyper flash memory.

3.1.5 Dependence command of the Hyper flash / Hyper RAM

The HYPB driver doesn't depend on the kind of the Hyper Flash / Hyper RAM memory. To support various Hyper Flash / Hyper RAM memories, the control commands, which depended on the Hyper Flash / Hyper RAM memory, must be set.

The command depending on the Hyper Flash / Hyper RAM memory sets it to r_hypb_Command_t structure and it must be handed to the Open function.

3.2 Using the API

3.2.1 Initialization / De-Initialization

R_HYPB_Init initializes the driver and the hardware as far as necessary. The Unit parameter holds a number that specifies the HYPB unit number being initialized. R_HYPB_DeInit function de-initializes the driver and the hardware as far as necessary.

3.2.2 Hyper RAM - External address space mode

The sample code of External address space mode for Hyper RAM is shown below. After setting completion, the Hyper RAM is assigned to linear memory space from 0x60000000 to 0x607FFFFF.

```
void SampleExternalAddressMode(void)
{
    r_hypb_Config_t config;
    uint32_t* read_pointer;
    uint32_t* write_pointer;
    uint32_t i;

    /* Init */
    R_HYPB_Init(LOC_HYPB_UNIT);

    /* Open */
    config.DeviceType = R_HYPB_DEVICE_RAM;
    config.OpeMode = R_HYPB_MODE_EX_SPACE;
    config.MemorySize = 8 * 1024 * 1024; /* Byte */
    config.SectorSize = 0;
    config.WriteAlign = 0;
    config.RelaxSize = 0;
    config.ParamSec.Size = 0;
    config.ParamSec.Num = 0;
    config.ParamSec.Pos = R_HYPB_PARAM_SEC_TOP;
    config.Command = &r_hypdb_RamCmdTbl;

    R_HYPB_Open(LOC_HYPB_UNIT, &config);

    /* Read & Write sample */
    read_pointer = (uint32_t*)0x60000000;
    write_pointer = (uint32_t*)0x60100000;
    for (i = 0; i < 100; i++)
    {
        write_pointer[i] = read_pointer[i];
    }
}
```

3.2.3 Hyper Flash - External address space mode

The sample code of External address space mode for Hyper Flash is shown below. After setting completion, the Hyper Flash is assigned to linear memory space from 0x60000000 to 0x60FFFFFFF.

```
void SampleExternalAddressMode(void)
{
    r_hypb_Config_t config;
    uint32_t* read_pointer;
    uint32_t read_data[100];
    uint32_t i;

    /* Init */
    R_HYPB_Init(LOC_HYPB_UNIT);

    /* Open */
    config.DeviceType = R_HYPB_DEVICE_FLASH;
    config.OpeMode = R_HYPB_MODE_EX_SPACE;
    config.MemorySize = 16 * 1024 * 1024; /* Byte */
    config.SectorSize = 256 * 1024; /* Byte */
    config.WriteAlign = 512; /* Byte */
    config.RelaxSize = 0;
    config.ParamSec.Size = 0;
    config.ParamSec.Num = 0;
    config.ParamSec.Pos = R_HYPB_PARAM_SEC_TOP;
    config.Command = &r_hypdb_FlashCmdTbl;

    R_HYPB_Open(LOC_HYPB_UNIT, &config);

    /* Read sample */
    read_pointer = (uint32_t*)0x60000000;
    for (i = 0; i < 100; i++)
    {
        read_data[i] = read_pointer[i];
    }
}
```

3.2.4 Hyper Flash - Manual mode

The sample code of manual mode for Hyper Flash is shown below.

```
void SampleManualMode(void)
{
    r_hypb_Config_t config;
    uint8_t data[100];
    uint32_t i;
    uint32_t word_size = 50;
    uint32_t word_addr = 0;

    /* Init */
    R_HYPB_Init(LOC_HYPB_UNIT);

    /* Open */
    config.DeviceType = R_HYPB_DEVICE_FLASH;
    config.OperMode = R_HYPB_MODE_MANUAL;
    config.MemorySize = 16 * 1024 * 1024; /* Byte */
    config.SectorSize = 256 * 1024; /* Byte */
    config.WriteAlign = 512; /* Byte */
    config.RelaxSize = 0;
    config.ParamSec.Size = 0;
    config.ParamSec.Num = 0;
    config.ParamSec.Pos = R_HYPB_PARAM_SEC_TOP;
    config.Command = &r_hypdb_FlashCmdTbl;

    R_HYPB_Open(LOC_HYPB_UNIT, &config);

    /* Read data */
    R_HYPB_DataRead(LOC_HYPB_UNIT, word_addr, data, word_size);
    /* Make write data */
    for (i = 0; i < 100; i++) {
        data[i]++;
    }
    /* Unprotect */
    R_HYPB_DataProtect(LOC_HYPB_UNIT, word_addr, word_size, R_HYPB_MODE_UNPROTECT);
    /* Erase sector */
    R_HYPB_DataErase(LOC_HYPB_UNIT, word_addr, word_size, R_NULL);
    /* Write data */
    R_HYPB_DataWrite(LOC_HYPB_UNIT, word_addr, data, word_size, R_NULL);
    /* Protect */
    R_HYPB_DataProtect(LOC_HYPB_UNIT, word_addr, word_size, R_HYPB_MODE_PROTECT);
}
```

3.3 Device difference

The following table shows the function differences depending on the device.

Table 3-2 APIs supported by HYPB driver

Feature	RH850/D1x Device Name			
	D1L2(H)	D1M1(H)	D1M1-V2, D1M1A	D1M2(H)
All API of HYPB driver	No	No	Full	No

3.4 Header File List

Table 3-3 Header File List

No.	Header File Name	Description
(1)	r_hypb_api.h	Header file for HYPB API.
(2)	r_typedefs.h	Header file for predefined data types.

4.Functions

4.1 Function List

This section describes about the HYPB API functions which are in [Table 4-1](#) and executable state of each function is described in the specification of each function.

Table 4-1 List of HYPB API Functions

Function Name	Purpose
<i>R_HYPB_Init</i>	This function initializes the HYPB driver.
<i>R_HYPB_DeInit</i>	This function de-initializes the HYPB driver.
<i>R_HYPB_Open</i>	This function opens the HYPB driver.
<i>R_HYPB_Close</i>	This function closes the HYPB driver.
<i>R_HYPB_DataProtect</i>	This function sets the protection mode of the hyper RAM/flash memory.
<i>R_HYPB_DataErase</i>	This function erases the data in the hyper RAM/flash memory.
<i>R_HYPB_DataWrite</i>	This function writes data to the hyper RAM/flash memory.
<i>R_HYPB_DataRead</i>	This function reads data from the hyper RAM/flash memory.
<i>R_HYPB_UserCmdIssue</i>	This function executes user command sequence.
<i>R_HYPB_VersionStringGet</i>	This function returns the version string of this HYPB driver.
<i>R_HYPB_MacroVersionGet</i>	This function returns the major and minor version of the H/W macro.

4.2 HYPB API Functions

This chapter describes the application interface functions, which are required for general use of the driver.

4.2.1 Basic functions

The section describes driver functions, which are required for general use of the driver, but which are related to a specific functionality of the macro itself.

4.2.1.1 R_HYPB_Init

Function Prototypes

```
r_hypb_Error_t R_HYPB_Init(const unit32_t    Unit)
```

Input Parameter

Table 4-2 Input parameter of R_HYPB_Init

Parameter	Description
Unit	Specifies the HYPB unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_DEVICE	- HYPB driver is not applicable to target device.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.

Description

This function initializes the HYPB driver.

This function calls R_HYPB_Sys_Init to initialize environment-dependent setting.

If the function successfully executes, the return code will be R_OCTA_ERR_OK and the state will be in the Initialize state.

Reentrancy

Non-reentrant

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t

4.2.1.2 R_HYPB_DeInit**Function Prototypes**

```
r_hypb_Error_t R_HYPB_DeInit(const uint32_t      Unit)
```

Input Parameter**Table 4-3 Input parameter of R_HYPB_DeInit**

Parameter	Description
Unit	Specifies the HYPB unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.

Description

This function de-initializes the HYPB driver.

This function calls R_HYPB_Sys_DeInit to de-initialize environment-dependent setting.

If the function successfully executes, the return code will be R_OCTA_ERR_OK and the state will be in the Uninitialize state.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t

4.2.1.3 R_HYPB_Open**Function Prototypes**

```
r_hypb_Error_t R_HYPB_Open(const uint32_t          Unit,  
                           const r_hypb_Config_t * const Config)
```

Input Parameter**Table 4-4 Input parameter of R_HYPB_Open**

Parameter	Description
Unit	Specifies the HYPB unit number.
Config	Pointer to the r_hypb_Config_t structure.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_HYPB_ERR_OK	- No error has occurred.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_HYPB_ERR_RANGE_PARAM	- A parameter is out of range.
R_HYPB_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.
R_HYPB_ERR_LATENCY	- A latency value is invalid.

Description

This function opens the HYPB driver.

This function will access the Configuration register to set latency code and parameter-sector information. This process is dependent on R_HYPB_Sys_ConfigRegSet.

At the first access, ReadCfg1 member of r_hypb_Command_t structure and WriteCfg1 member of r_hypb_Command_t structure are used to access to Configuration register.

After the second time, ReadCfg2 member of r_hypb_Command_t structure and WriteCfg2 member of r_hypb_Command_t structure are used.

If ReadCfg1/ReadCfg2 or WriteCfg1/WriteCfg2 is not supported, this function dose not update the Configuration register. In this case, this function assumes that latency code and parameter-sector set by argument have already set to register.

After Configuration register is updated, in case of manual mode, this function executes Open member of r_hypb_Command_t structure command if it is supported.

In case of external address space mode, Read member of r_hypb_Command_t structure command is set to H/W and it enables access from external address space.

When R_HYPB_MODE_EX_SPACE is specified, this function will fail and return R_HYPB_ERR_COMMAND or R_HYPB_ERR_LATENCY under following conditions.

- Read member of r_hypb_Command_t structure command is not supported.
- Read member of r_hypb_Command_t structure transaction cycle is not 1.
- Read member of r_hypb_Command_t structure latency cycles are less than 8.
- Read member of r_hypb_Command_t structure latency cycles are not even value.

If the function successfully executes, the return code will be R_HYPB_ERR_OK. And the status will be changed to Idle if R_HYPB_MODE_MANUAL is specified. The status will be changed to Executing if R_HYPB_MODE_EX_SPACE is specified.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t
r_hypb_Config_t

4.2.1.4 R_HYPB_Close**Function Prototypes**

```
r_hypb_Error_t R_HYPB_Close( const uint32_t Unit)
```

Input Parameter**Table 4-5 Input parameter of R_HYPB_Close**

Parameter	Description
Unit	Specifies the HYPB unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function closes the HYPB driver.

In case of manual mode, this function executes Close member of r_hypb_Command_t structure command if it is supported.

In case of external address space mode, this function disables access from the external space.

If the function successfully executes, the return code will be R_HYPB_ERR_OK and the state will be in the Initialize state.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t

4.2.1.5 R_HYPB_DataProtect

Function Prototypes

```
r_hypb_Error_t R_HYPB_DataProtect(const uint32_t Unit,
                                   const uint32_t WordAddr,
                                   const uint32_t WordSize,
                                   const r_hypb_ProtectionMode_t Mode)
```

Input Parameter

Table 4-6 Input parameter of R_HYPB_DataProtect

Parameter	Description
Unit	Specifies the HYPB unit number.
WordAddr	Protect control start Word-address of the hyper flash memory.
WordSize	Data Word-size to erase.
Mode	Protection mode.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter is incorrect.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_HYPB_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_HYPB_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.

Description

This function sets the protection mode of the hyper flash memory.

This function executes Protect member of r_hypb_Command_t structure command.

This function assumes that Protect command can be set protection mode for each sector (or parameter-sector).

This function controls sectors in following range.

Start sector Sector (or parameter-sector) that WordAddr is belonged.

End sector Sector (or parameter-sector) that (WordAddr + WordSize - 1) is belonged.

This function executes Protect command without checking the protecting status of sector.

The write or erase command to hyper flash memory will fail if target sector is protected.

This function will fail and return R_HYPB_ERR_COMMAND under following conditions.

- SectorSize member of r_hypb_Config_t structure = 0.
- Protect member of r_hypb_Command_t structure command is not supported.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t
r_hypb_ProtectionMode_t

4.2.1.6 R_HYPB_DataErase

Function Prototypes

```
r_hypb_Error_t R_HYPB_DataErase(const uint32_t      Unit,
                                const uint32_t      WordAddr,
                                const uint32_t      WordSize,
                                uint32_t* const ResultSize )
```

Input Parameter

Table 4-7 Input parameter of R_HYPB_DataErase

Parameter	Description
Unit	Specifies the HYPB unit number.
WordAddr	Erase start Word-address of the hyper flash memory.
WordSize	Data Word-size to erase.

Input-Output Parameter

None

Output Parameter

Table 4-8 Output parameter of R_HYPB_DataErase

Parameter	Description
ResultSize	Pointer to variable that receives the Word-size erased by this function. If not required, please set to R_NULL.

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_HYPB_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.
R_HYPB_ERR_PROTECTED	- A process is aborted because of memory protection.
R_HYPB_ERR_ABORTED	- A process is aborted because of other reason.
R_HYPB_ERR_TIMEOUT	- Status polling is timeout.

Description

This function erases the data in the hyper flash memory.

This function executes Erase member of `r_hypb_Command_t` structure command.

This function assumes that Erase command can be erased in a unit of sector (or parameter-sector).

This function erases the sectors in following range.

Start sector Sector (or parameter-sector) that WordAddr is belonged.

End sector Sector (or parameter-sector) that (WordAddr + WordSize - 1) is belonged.

This function will fail and return `R_HYPB_ERR_COMMAND` under following conditions.

- SectorSize member of `r_hypb_Config_t` structure = 0.
- Erase member of `r_hypb_Command_t` structure command is not supported.
- Multiple sectors are specified and ReadSts member of `r_hypb_Command_t` structure command is not supported.

[Wait mode]

If ReadSts member of `r_hypb_Command_t` structure command is supported, this function waits to finish the erasing process and checks the result.

Actual erased size is stored to ResultSize.

Result check process is dependent on `R_HYPB_Sys_StatusRegCheck`.

[No-Wait mode]

If ReadSts member of `r_hypb_Command_t` structure command is not supported, this function does not wait to finish the erasing process and dose not check the result.

After one Erase command sequence is finished, this function will return `R_HYPB_ERR_OK` and the size that is set to command is stored to ResultSize.

If the function successfully executes, the return code will be `R_HYPB_ERR_OK`.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- `R_HYPB_Sys_Lock`
- `R_HYPB_Sys_Unlock`

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

`r_hypb_Error_t`

4.2.1.7 R_HYPB_DataWrite

Function Prototypes

```
r_hypb_Error_t R_HYPB_DataWrite(const uint32_t      Unit,
                                const uint32_t      WordAddr,
                                const uint8_t*       Buf,
                                const uint32_t      WordSize,
                                uint32_t* const ResultSize)
```

Input Parameter

Table 4-9 Input parameter of R_HYPB_DataWrite

Parameter	Description
Unit	Specifies the HYPB unit number.
WordAddr	The parameter specifies the write Word-address of the hyper flash memory.
Buf	This is a pointer to the buffer stored write data.
Mode	The parameter specifies the protection mode of the serial flash memory.

Input-Output Parameter

None

Output Parameter

Table 4-10 Output parameter of R_HYPB_DataWrite

Parameter	Description
ResultSize	Pointer to variable that receives the Word-size written by this function. If not required, please set to R_NULL.

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_HYPB_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.
R_HYPB_ERR_PROTECTED	- A process is aborted because of memory protection.
R_HYPB_ERR_ABORTED	- A process is aborted because of other reason.
R_HYPB_ERR_TIMEOUT	- Status polling is timeout

Description

This function writes data to the hyper flash memory.

In order to write data to Hyper Flash, the data of the sector must have been erased previously.

This function executes WriteBufFlash member of `r_hypb_Command_t` structure or Write member of `r_hypb_Command_t` structure command under following condition.

- If Write command is supported, Write command is used only.
- If Write command is not supported, WriteBufFlash command is used only.
- If both commands are not supported, this function will fail and return `R_HYPB_ERR_COMMAND`.

When Write command is used and Write command doesn't have `R_HYPB_CMD_WRITE_A` option, this function has the possibility that the processing takes time. Therefore, `R_HYPB_Sys_Relax` is executed in units of `RelaxSize` member of `r_hypb_Config_t` structure.

[Wait mode]

If `ReadSts` member of `r_hypb_Command_t` structure command is supported, this function waits to finish the writing process and checks the result.

Actual written size is stored to `ResultSize`.

Result check process is dependent on `R_HYPB_Sys_StatusRegCheck`.

[No-Wait mode]

If `ReadSts` member of `r_hypb_Command_t` structure command is not supported, this function does not wait to finish the writing process and does not check the result.

If WriteBufFlash command is used, this function will return `R_HYPB_ERR_OK` after only one WriteBufFlash command sequence is issued.

If Write command is used and Write command has `R_HYPB_CMD_WRITE_A` option, this function will return `R_HYPB_ERR_OK` after only one Write command sequence is issued.

If Write command is used and Write command doesn't have `R_HYPB_CMD_WRITE_A` option, Write command is executed continuously without status check.

The size that is set to command is stored to `ResultSize`.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- `R_HYPB_Sys_Lock`
- `R_HYPB_Sys_Unlock`

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

`r_hypb_Error_t`

4.2.1.8 R_HYPB_DataRead

Function Prototypes

```
r_hypb_Error_t R_HYPB_DataRead(const uint32_t      Unit,
                                const uint32_t      WordAddr,
                                uint8_t* const Buf,
                                const uint32_t      WordSize)
```

Input Parameter

Table 4-11 Input parameter of R_HYPB_DataRead

Parameter	Description
Unit	Specifies the HYPB unit number.
WordAddr	The parameter specifies the read Word-address of the hyper flash memory.
WordSize	The parameter specifies the data Word-size to read.

Input-Output Parameter

None

Output Parameter

Table 4-12 Output parameter of R_HYPB_DataRead

Parameter	Description
Buf	This is a pointer to the buffer to store the read data.

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.

Description

This function reads data from the hyper flash memory.

This function will fail and return R_HYPB_ERR_COMMAND under following conditions.

- Read member of r_hypb_Command_t structure command is not supported.

This function has the possibility that the processing takes time. Therefore, R_HYPB_Sys_Relax is executed in units of RelaxSize member of r_hypb_Config_t structure.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t

CONFIDENTIAL

4.2.1.9 R_HYPB_UserCmdIssue

Function Prototypes

```
r_hypb_Error_t R_HYPB_UserCmdIssue(const uint32_t Unit,  
                                     const r_hypb_CmdSequence_t* CmdSequence,  
                                     uint8_t * Buf,  
                                     const uint32_t BufSize)
```

Input Parameter

Table 4-13 Input parameter of R_HYPB_UserCmdIssue

Parameter	Description
Unit	Specifies the HYPB unit number.
CmdSequence	Command sequence.
BufSize	Byte-size of data that can be stored to Buf.

Input-Output Parameter

None

Output Parameter

Table 4-14 Output parameter of R_HYPB_UserCmdIssue

Parameter	Description
Buf	This is a pointer to the Buffer to store the read data. If read transaction is not existed in command sequence, please set to R_NULL.

Return Codes

R_HYPB_ERR_OK	- No error occurred.
R_HYPB_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_HYPB_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_HYPB_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_HYPB_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_HYPB_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	- A command is not supported.

Description

This function executes user command sequence.

Following Command Option are available.

- R_HYPB_CMD_WRITE16
- R_HYPB_CMD_READ16
- R_HYPB_CMD_READ32
- R_HYPB_CMD_READ64

If read transaction is existed in command sequence, read data will be stored to Buf. Read size is 16bit, 32bit or 64bit. It depends on Command Option. See [Table 5-3](#). If BufSize is less than Read size, specified size by BufSize is stored to Buf.

e.g. When R_HYPB_CMD_READ64 is specified and BufSize = 8, stored data is as follows.

```
Buf[0] = 1st WordData[15:08]
Buf[1] = 1st WordData[07:00]
:
Buf[6] = 4th WordData[15:08]
Buf[7] = 4th WordData[07:00]
```

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_HYPB_Sys_Lock
- R_HYPB_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t
r_hypb_CmdSequence_t

4.2.1.10 R_HYPB_VersionStringGet

Function Prototypes

```
const uint8_t* R_HYPB_VersionStringGet(void)
```

Input Parameter

None

Input-Output Parameter

None

Output Parameter

None

Return Codes

Version string.

Description

This function returns version string of the HYPB driver.

Reentrancy

Reentrant.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

None

4.2.1.11 R_HYPB_MacroVersionGet**Function Prototypes**

```
r_hypb_Error_t R_HYPB_MacroVersionGet(uint32_t * const Major,  
                                       uint32_t * const Minor)
```

Input Parameter

None

Input -Output Parameter

None

Output Parameter**Table 4-15 Output parameter of R_HYPB_MacroVersionGet**

Parameter	Description
Major	The major version.
Minor	The minor version.

Return Codes

R_HYPB_ERR_OK	- No error has occurred.
R_HYPB_ERR_PARAM_INCORRECT	- Either parameter Major or parameter Minor was R_NULL

Description

This function returns the major and minor version of the H/W macro.

Reentrancy

Reentrant.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about HYPB unit status conditions.

See also

r_hypb_Error_t

4.2.2 Interrupt functions

None.

5.Types

5.1 Basic Types

This section shows the basic types used on this library.

Table 5-1 Basic type

Types	Definition		Basic types
char_t	typedef char	char_t	signed char
int8_t	typedef signed char	int8_t	signed char
int16_t	typedef signed short	int16_t	signed short
int32_t	typedef signed int	int32_t	signed int
int64_t	typedef signed long long	int64_t	signed long long
uint8_t	typedef unsigned char	uint8_t	unsigned char
uint16_t	typedef unsigned short	uint16_t	unsigned short
uint32_t	typedef unsigned int	uint32_t	unsigned int
uint64_t	typedef unsigned long long	uint64_t	unsigned long long
float32_t	typedef float	float32_t	float
float64_t	typedef double	float64_t	double

5.2 Definition

This section shows the definitions used in HYPB API.

Table 5-2 Definition of HYPB API

Name	Description
R_HYPB_VERSION_HI	MSB byte of the version information. It is major version information. This value is changed with release version.
R_HYPB_VERSION_LO	LSB byte of the version information. It is minor version information. This value is changed with release version.

Table 5-3 Definition of Command Option

Name	Description
R_HYPB_CMD_WRITE	Use Write transaction.
R_HYPB_CMD_READ	Use Read transaction.
R_HYPB_CMD_SIZE_16	Size is 16 bit.
R_HYPB_CMD_SIZE_32	Size is 32 bit.
R_HYPB_CMD_SIZE_64	Size is 64 bit.
R_HYPB_CMD_SIZE_MASK	Mask for size get.
R_HYPB_CMD_APPROPRIATE	Size is appropriate (16/32/64) bit.
R_HYPB_CMD_WRITE16	Use Write transaction. Write size is 16 bit. It is default behavior
R_HYPB_CMD_WRITE_A	Use Write transaction. Write size is appropriate (16/32/64) bit. This flag can be set with R_HYPB_CMD_SET_DATA
R_HYPB_CMD_READ16	Use Read transaction. Read size is 16 bit.
R_HYPB_CMD_READ32	Use Read transaction. Read size is 32 bit.
R_HYPB_CMD_READ64	Use Read transaction. Read size is 64 bit.
R_HYPB_CMD_SET_ADDR	Update Address member to Word-address.
R_HYPB_CMD_SET_SA	Update Address member to sector address. This flag can be set when SectorSize member of r_hypb_Config_t structure is not 0.
R_HYPB_CMD_SET_DATA	Update WriteData member to write data.
R_HYPB_CMD_SET_COUNT	Update WriteData member to loop count.
R_HYPB_CMD_LOOP	The transaction is executed repeatedly.

5.3 Enumerated Type

This section shows the enumerated types used in HYPB API Function.

5.3.1 r_hypb_Error_t

Description

HYPB driver error code.

If an error occurs, these enumerations give information about the reason.

Definition

```
typedef enum
{
    R_HYPB_ERR_OK = 0,
    R_HYPB_ERR_NG,
    R_HYPB_ERR_PARAM_INCORRECT,
    R_HYPB_ERR_RANGE_UNIT,
    R_HYPB_ERR_RANGE_PARAM,
    R_HYPB_ERR_NOT_ACCEPTABLE,
    R_HYPB_ERR_DEVICE,
    R_HYPB_ERR_FATAL_OS,
    R_HYPB_ERR_FATAL_HW,
    R_HYPB_ERR_COMMAND,
    R_HYPB_ERR_LATENCY,
    R_HYPB_ERR_PROTECTED,
    R_HYPB_ERR_ABORTED,
    R_HYPB_ERR_TIMEOUT
} r_hypb_Error_t;
```

Table 5-4 Enumerator of r_hypb_Error_t

Name	Description
R_HYPB_ERR_OK	No error occurred.
R_HYPB_ERR_NG	An error has occurred, but no specific error code is defined for it.
R_HYPB_ERR_PARAM_INCORRECT	A parameter provided to a function was incorrect.
R_HYPB_ERR_RANGE_UNIT	The unit-number was outside the range.
R_HYPB_ERR_RANGE_PARAM	Parameter is the outside the range.
R_HYPB_ERR_NOT_ACCEPTABLE	A function was called in an incorrect state.
R_HYPB_ERR_DEVICE	HYPB driver is not applicable to target d1x device.
R_HYPB_ERR_FATAL_OS	Fatal error has occurred at OS interface.
R_HYPB_ERR_FATAL_HW	Fatal error has occurred at H/W.
R_HYPB_ERR_COMMAND	A command is not supported.
R_HYPB_ERR_LATENCY	A latency value is invalid.
R_HYPB_ERR_PROTECTED	A process is aborted because of memory protection.
R_HYPB_ERR_ABORTED	A process is aborted because of other reason.
R_HYPB_ERR_TIMEOUT	Status polling is timeout.

See also

None

5.3.2 r_hypb_DeviceType_t**Description**

This type describes the control device type.

Definition

```
typedef enum
{
    R_HYPB_DEVICE_FLASH = 0,
    R_HYPB_DEVICE_RAM
} r_hypb_DeviceType_t;
```

Table 5-5 Enumerator of r_hypb_DeviceType_t

Name	Description
R_HYPB_DEVICE_FLASH	Hyper Flash.
R_HYPB_DEVICE_RAM	Hyper RAM. Hyper RAM should consist of single die.

See also

None

5.3.3 r_hypb_OperatingMode_t

Description

This type describes the operating mode.

Definition

```
typedef enum
{
    R_HYPB_MODE_EX_SPACE = 0,
    R_HYPB_MODE_MANUAL
} r_hypb_OperatingMode_t;
```

Table 5-6 Enumerator of r_hypb_OperatingMode_t

Name	Description
R_HYPB_MODE_EX_SPACE	External address space mode. In case of Hyper Flash, only Read memory access is possible. In case of Hyper RAM, Read/Write memory access is possible.
R_HYPB_MODE_MANUAL	Manual mode. In case of Hyper Flash, Read/Write access to register/memory area is possible. In case of Hyper RAM, Read/Write access to register area is possible.

See also

None

5.3.4 r_hypb_ParamSecPos_t**Description**

This type describes the position of parameter-sector.

Definition

```
typedef enum
{
    R_HYPB_PARAM_SEC_TOP      = 0,
    R_HYPB_PARAM_SEC_BOTTOM
} r_hypb_ParamSecPos_t;
```

Table 5-7 Enumerator of r_hypb_ParamSecPos_t

Name	Description
R_HYPB_PARAM_SEC_TOP	Parameter-sector is mapped into the top of the first sector of Hyper RAM/flash memory.
R_HYPB_PARAM_SEC_BOTTOM	Parameter-sector is mapped into the bottom of the last sector of Hyper RAM/flash memory.

See also

None

5.3.5 r_hypb_ProtectionMode_t**Description**

This type describes the protection mode of the hyper RAM/flash memory.

Definition

```
typedef enum
{
    R_HYPB_MODE_PROTECT = 0,
    R_HYPB_MODE_UNPROTECT
} r_hypb_ProtectionMode_t;
```

Table 5-8 Enumerator of r_hypb_ProtectionMode_t

Name	Description
R_HYPB_MODE_PROTECT	Protection mode.
R_HYPB_MODE_UNPROTECT	Un-protection mode.

See also

None

5.3.6 r_hypb_CmdType_t

Description

This type describes the command type that is defined r_hypb_Command_t.

Definition

```
typedef enum
{
    R_HYPB_CMD_TYPE_OPEN,
    R_HYPB_CMD_TYPE_CLOSE,
    R_HYPB_CMD_TYPE_READ,
    R_HYPB_CMD_TYPE_WRITE,
    R_HYPB_CMD_TYPE_WRITE_BUF_FLASH,
    R_HYPB_CMD_TYPE_ERASE,
    R_HYPB_CMD_TYPE_PROTECT,
    R_HYPB_CMD_TYPE_READ_CFG_1,
    R_HYPB_CMD_TYPE_READ_CFG_2,
    R_HYPB_CMD_TYPE_WRITE_CFG_1,
    R_HYPB_CMD_TYPE_WRITE_CFG_2,
    R_HYPB_CMD_TYPE_READ_STS,
    R_HYPB_CMD_TYPE_USER,
    R_HYPB_CMD_TYPE_MAX
} r_hypb_CmdType_t;
```

Table 5-9 Enumerator of r_hypb_CmdType_t

Name	Description
R_HYPB_CMD_TYPE_OPEN	Command of Open.
R_HYPB_CMD_TYPE_CLOSE	Command of Close.
R_HYPB_CMD_TYPE_READ	Command of Read.
R_HYPB_CMD_TYPE_WRITE	Command of Write.
R_HYPB_CMD_TYPE_WRITE_BUF_FLASH	Command of WriteBufFlash.
R_HYPB_CMD_TYPE_ERASE	Command of Erase.
R_HYPB_CMD_TYPE_PROTECT	Command of Protect.
R_HYPB_CMD_TYPE_READ_CFG_1	Command of ReadCfg1.
R_HYPB_CMD_TYPE_READ_CFG_2	Command of ReadCfg2.
R_HYPB_CMD_TYPE_WRITE_CFG_1	Command of WriteCfg1.
R_HYPB_CMD_TYPE_WRITE_CFG_2	Command of WriteCfg2.
R_HYPB_CMD_TYPE_READ_STS	Command of ReadSts.
R_HYPB_CMD_TYPE_USER	Command of user defined.
R_HYPB_CMD_TYPE_MAX	The number of defined Command.

See also

None

5.4 Structure Type

This section shows the structure used in HYPB API Function.

5.4.1 r_hypb_ParamSec_t

Description

This type describes the parameter-sector information.

Example:

MemorySize member of r_hypb_Config_t structure =16*1024*1024,

SectorSize member of r_hypb_Config_t structure =256*1024

(1) Size=4096, Num=8, Pos=R_HYPB_PARAM_SEC_TOP

Word-Address	Sector Number
00000000h~000007FFh	Parameter-Sector #0
00000800h~00000FFFh	Parameter-Sector #1
:	:
00003800h~00003FFFh	Parameter-Sector #7
00004000h~0001FFFFh	Normal Sector #0
00020000h~0003FFFFh	Normal Sector #1
:	:

(2) Size=4096, Num=8, Pos=R_HYPB_PARAM_SEC_BOTTOM

Word-Address	Sector Number
:	:
007C0000h~007DFFFFh	Normal Sector #62
007E0000h~007FBFFFh	Normal Sector #63
007FC000h~007FC7FFh	Parameter-Sector #0
007FC800h~007FCFFFh	Parameter-Sector #1
:	:
007C0000h~007DFFFFh	Parameter-Sector #7

(3) ParamSec member of r_hypb_Config_t structure =R_NULL

Word-Address	Sector Number
00000000h~0001FFFFh	Normal Sector #0
00020000h~0003FFFFh	Normal Sector #1
:	:
007C0000h~007DFFFFh	Normal Sector #62
007E0000h~007FBFFFh	Normal Sector #63

Definition

```
typedef struct
{
    uint32_t          Size;
    uint32_t          Num;
    r_hypb_ParamSecPos_t Pos;
} r_hypb_ParamSec_t;
```

Table 5-10 Member of r_hypb_ParamSec_t

Name	Description
Size	Size of one parameter-sector (Byte). This size must be power-of-two value (2^n). If parameter sector is not defined or not used, please set to 0.
Num	The number of parameter-sectors. The total size of parameter-sector (Size * Num) must be less than SectorSize member of r_hypb_Config_t structure. If parameter sector is not defined or not used, please set to 0.
Pos	Position of parameter-sectors.

See also

r_hypb_ParamSecPos_t

5.4.2 r_hypb_CmdTransaction_t**Description**

This type describes the hyper RAM/flash commands.

Definition

```
typedef struct
{
    uint32_t      Address;
    uint16_t      WriteData;
    uint8_t       CmdFlags;
    uint8_t       Latency;
    uint32_t      OpeFlags;
} r_hypb_CmdTransaction_t;
```

Table 5-11 Member of r_hypb_CmdTransaction_t

Name	Description
Address	Address data to set CA[44:16] and CA[02:00].
WriteData	Write data (only supports a single write transaction of 16 bits.)
CmdFlags	Flags data to set CA[47:45]. Please set to MSB 3bit.
Latency	Latency in units of clock cycles. Latency is from CA[23:16] access timing to data Read/Write timing. If latency is needed, the range is 3 to 20. If no latency is needed, please set to 0.
OpeFlags	Operation Flags. Multiple flags can be set with ' '. See Table 5-3 .

See also

None

5.4.3 r_hypb_CmdSequence_t**Description**

This type describes the hyper RAM/flash commands.

Definition

```
typedef struct
{
    r_hypb_CmdTransaction_t*    Transaction;
    uint8_t                     Cycle;
} r_hypb_CmdSequence_t;
```

Table 5-12 Member of r_hypb_CmdSequence_t

Name	Description
Transaction	Pointer to Command Transaction Array. This memory must be kept allocating till R_HYPB_Close is finished. If the command is not supported, please set to R_NULL.
Cycle	Number of Command Transaction Array. If the command is not supported, please set to 0.

See also

r_hypb_CmdTransaction_t

5.4.4 r_hypb_Latency_t**Description**

This type describes the latency.

Definition

```
typedef struct
{
    uint8_t      Code;
    uint8_t      ClkDelay;
    uint8_t      NagDelay;
    uint8_t      NextAcc;
} r_hypb_Latency_t;
```

Table 5-13 Member of r_hypb_Latency_t

Name	Description
Code	Initial latency code to set to Configuration register.
ClkDelay	Sets the period from MCS pin assertion to MCK oscillation. Range is 1 to 8 (clock). (e.g. It is described as tCSS in data sheet of IS26KL128S.)
NagDelay	Sets the period from the time the last MCK edge is sent of a transfer to MCS pin negation. Range is 1 to 8. Actual Delay is set value + 0.5 clk. If NagDelay=3 is set, actual delay is 3.5 clock. (e.g. It is described as tCSH in data sheet of IS26KL128S.)
NextAcc	Sets the period from transaction end to next transaction start. Range is 1 to 8 (clock). (e.g. It is described as tCSHI in data sheet of IS26KL128S.)

See also

None

5.4.5 r_hypb_Command_t

Description

This type describes the hyper RAM/flash commands.

Definition

```
typedef struct
{
    r_hypb_Latency_t           Latency;
    r_hypb_CmdSequence_t      Open;
    r_hypb_CmdSequence_t      Close;
    r_hypb_CmdSequence_t      Read;
    r_hypb_CmdSequence_t      Write;
    r_hypb_CmdSequence_t      WriteBufFlash;
    r_hypb_CmdSequence_t      Erase;
    r_hypb_CmdSequence_t      Protect;
    r_hypb_CmdSequence_t      ReadCfg1;
    r_hypb_CmdSequence_t      ReadCfg2;
    r_hypb_CmdSequence_t      WriteCfg1;
    r_hypb_CmdSequence_t      WriteCfg2;
    r_hypb_CmdSequence_t      ReadSts;
} r_hypb_Command_t;
```


Table 5-14 Member of r_hypb_Command_t

Name	Description
Latency	Latency setting.
Open	Command sequence of Open. Following Operation Flags are available. R_HYPB_CMD_WRITE16
Close	Command sequence of Close. Following Operation Flags are available. R_HYPB_CMD_WRITE16
Read	Command sequence of Read. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_READ16 / R_HYPB_CMD_READ32 / R_HYPB_CMD_READ64 / R_HYPB_CMD_SET_ADDR
Write	Command sequence of Write directly. This command is used in case of manual mode. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_WRITE_A / R_HYPB_CMD_SET_ADDR / R_HYPB_CMD_SET_DATA
WriteBufFlash	Command sequence of Write via write Buffer. This command is used in case of manual mode. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_SET_ADDR / R_HYPB_CMD_SET_SA / R_HYPB_CMD_SET_DATA / R_HYPB_CMD_SET_COUNT / R_HYPB_CMD_LOOP
Erase	Command sequence of Sector Erase. This command is used in case of manual mode. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_SET_ADDR / R_HYPB_CMD_SET_SA
Protect	Command sequence of Sector Protect. This command is used in case of manual mode. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_SET_ADDR / R_HYPB_CMD_SET_SA / R_HYPB_CMD_SET_DATA
ReadCfg1	Command sequence of Read Configuration Register at the first time. This command is used before initial latency code is set. Please take care in latency setting. (e.g. Latency is written in Non-Volatile Configuration register in case of IS26KL128S.) Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_READ16
ReadCfg2	Command sequence of Read Configuration Register after the second time. This command is used after initial latency code is set. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_READ16
WriteCfg1	Command sequence of Write Configuration Register at the first time. This command is used when initial latency code is set. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_SET_DATA
WriteCfg2	Command sequence of Write Configuration Register after the second time. This command is used after initial latency code is set. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_SET_DATA
ReadSts	Command sequence of Read status Register. This command is used in case of manual mode. Following Operation Flags are available. R_HYPB_CMD_WRITE16 / R_HYPB_CMD_READ16

See also

r_hypb_CmdSequence_t
r_hypb_Latency_t

5.4.6 r_hypb_Config_t

Description

This type describes the configuration of the unit.

Definition

```
typedef struct
{
    r_hypb_DeviceType_t      DeviceType;
    r_hypb_OperatingMode_t   OpeMode;
    uint32_t                 MemorySize;
    uint32_t                 SectorSize;
    uint32_t                 WriteAlign;
    uint32_t                 RelaxSize;
    r_hypb_ParamSec_t        ParamSec;
    r_hypb_Command_t*        Command;
} r_hypb_Config_t;
```

Table 5-15 Member of r_hypb_Config_t

Name	Description
DeviceType	Control device type.
OpeMode	Operating mode.
MemorySize	Total memory byte-size of connected hyper RAM/flash memory. MemorySize should be multiple of SectorSize if SectorSize is defined. (e.g. This size is 16 MBytes(16*1024*1024), when IS26KL128S is connected.)
SectorSize	Sector byte-size of connected hyper RAM/flash memory. Sector size must be power-of-two value (2^n). This size is used in case of manual mode. This size is used when R_HYPB_CMD_SET_SA flag is included in the command. If Sector is not defined, please set to 0. (e.g. This size is 256 Kbytes when IS26KL128S is connected.)
WriteAlign	Alignment byte-address for WriteBufFlash command. Write align size must be power-of-two value (2^n). The range is 8 to 2048. This alignment is used in case of manual mode and it is used by R_HYPB_DataWrite. Write command sequence is closed in units of this alignment. If not required, please set to 0. (e.g. This size is 512 bytes (Write buffer line size) when IS26KL128S is connected.)
RelaxSize	Relax byte-size. Relax size must be multiple of 8-byte. This size is used in case of manual mode. Some APIs call R_HYPB_Sys_Relax in units of this size. If not required, please set to 0.
ParamSec	Parameter sector information. This information is used in case of manual mode.
Command	Command setting. This memory must be kept allocating till R_HYPB_Close is finished.

See also

[r_hypb_DeviceType_t](#)
[r_hypb_OperatingMode_t](#)
[r_hypb_ParamSec_t](#)
[r_hypb_Command_t](#)

Revision History	Renesas Graphics Library HyperBus Controller (HYPB) Driver User's Manual: Software
------------------	--

Rev.	Date	Description	
		Page	Summary
0.1	Nov 22, 2019	-	First edition.
1.0	April 24, 2020	-	Update Revision.

Renesas Graphics Library
HyperBus Controller (HYPB) Driver
User's Manual: Software

Publication Date: Rev.0.1 Nov 22, 2019
 Rev.1.0 April 24, 2020

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 Corporation

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc. Milpitas Campus

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.

Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus

6024 Silver Creek Valley Road, San Jose, CA 95138, USA

Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3

Tel: +1-905-237-2004

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany

Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China

Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China

Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong

Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan

Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, #06-02 Singapore 339949

Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia

Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India

Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea

Tel: +82-2-558-3737, Fax: +82-2-558-5338



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>

Renesas Graphics Library HyperBus Controller (HYPB) Driver



Renesas Electronics Corporation