RZ/A1H Group

Direct Memory Access Controller "DMAC_RM" Sample Driver (Attached to PFV)

R01AN1888EJ0103
Rev.1.03
Aug.24, 2017

Introduction

This application note describes the sample driver which uses the RZ/A1H's direct memory access controller (DMAC).

This program, which is attached to the PFV sample driver, controls the DMAC.

The DMAC_RM sample driver offers the following features:

- Starts and stops DMA in DMAC register mode or link mode.
- Uses the DMA extended resource selector as needed.

Target Device

RZ/A1H Group

RZ/A1M Group

RZ/A1L Group

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Table of contents

1. Specifications

Table 1-1 shows Peripheral Functions and Their Applications, and Figure 1-1 shows the Operation Overview.

Table 1-1 Peripheral Functions and Their Applications

Peripheral functions	Uses
Direct Memory Access Controller (DMAC) Ch.(0..15)	Direct memory access controller
Interrupt Controller (INTC)	DMAC interrupt control
Serial Communications Interface (SCIF)(UART)	Debug output

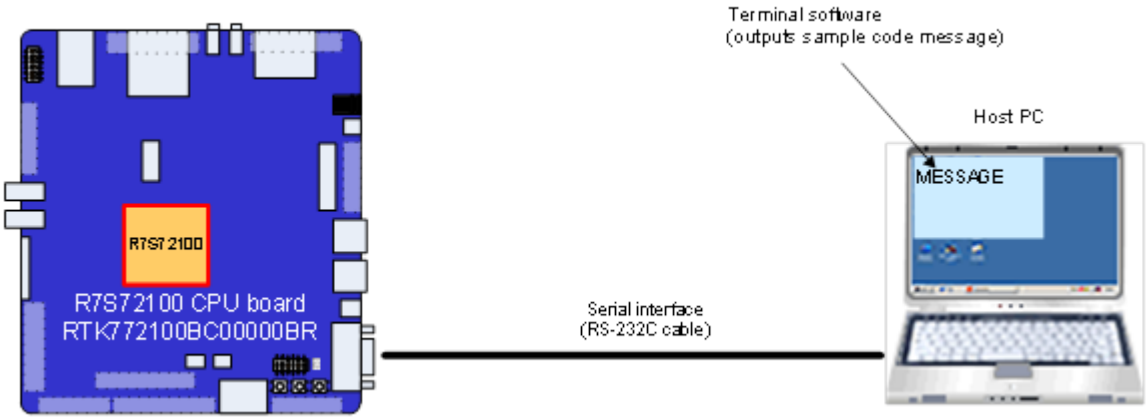


Figure 1-1 the Operation Overview

2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed below.

Table 2-1 Operation Check Conditions

Item		Description
MCU used		RZ/A1H
Operating frequency		CPU clock (Iϕ): 400MHz Image processing clock (Gϕ): 266.67MHz Internal bus clock (Bϕ): 133.33MHz Peripheral clock 1 (P1ϕ): 66.67MHz Peripheral clock 0 (P0ϕ): 33.33MHz
Operating voltage		Power supply voltage (I/O): 3.3V Power supply voltage (Internal): 1.18V
ARM	Integrated development environment	ARM® integrated development environment ARM Development Studio 5 (DS-5™) Version 5.16
	C compiler	ARM C/C++ Compiler/Linker/Assembler Ver.5.03 [Build 102]
IAR	Integrated development environment	IAR Embedded Workbench for ARM 7.80.4.12495
	C compiler	
Renesas	Integrated development environment	e2 studio (Version: 5.3.0.023)
	C compiler	GNUARM-NONE-EABI v16.01
Operating mode		Boot mode 0 (CS0 space 16bit boot)
Communication setting of terminal software		• Communication speed: 115200bps • Data length: 8 bits • Parity: None • Stop bit length: 1 bit • Flow control: None
Board used		GENMAI board • RTK772100BC00000BR (R7S72100 CPU board) • RTK77210000B00000BR (R7S72100 Option board)
Device used		Serial interface (D-sub 9-pin connector)

3. Reference Application Note(s)

For additional information associated with this document, refer to the following application note(s).

- RZ/A1H Example of Initialization (R01AN1646EJ)
- RZ/A1H Group I/O definition header file <iodefine.h> (R01AN1860EJ)
- RZ/A1H Group OS porting layer "OSPL" Sample Program (R01AN1887EJ)
- RZ/A1H Group Pixel Format Converter "PFV" Sample Driver (R01AN1880EJ)

4. Peripheral Functions

The basic function of the DMAC is described in the RZ/A1H Group User's Manual: Hardware.

5. Description of Hardware

5.1 Hardware Configuration

Figure 5-1 shows Examples of Hardware Devices Connected. Figure 5-2 shows Block Diagram.

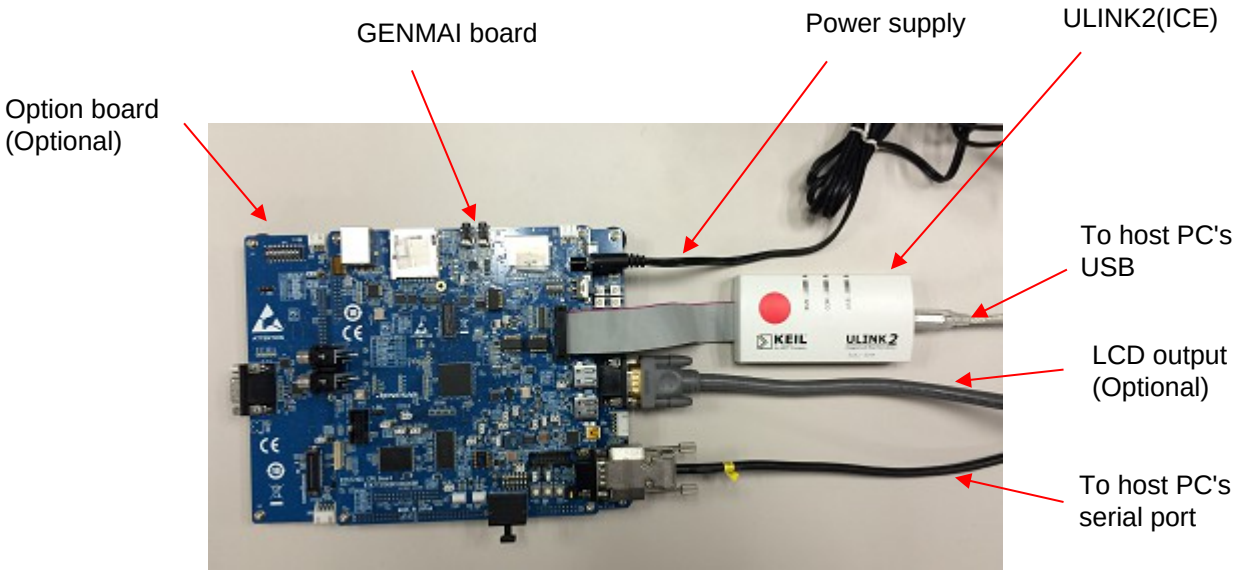


Figure 5-1 Examples of Hardware Devices Connected

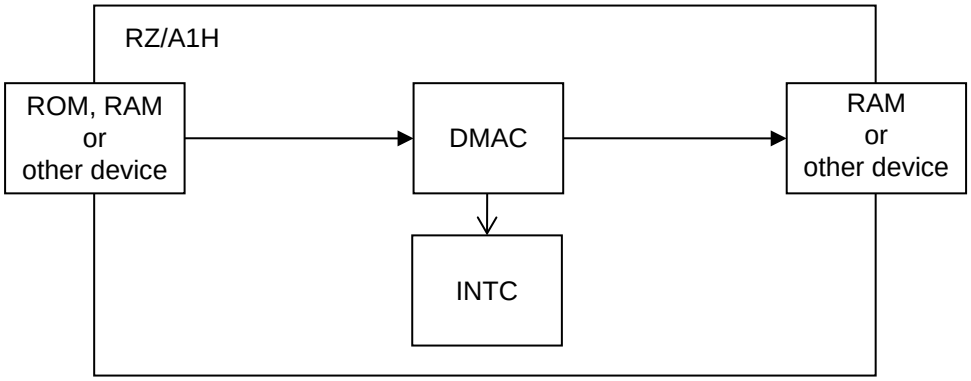


Figure 5-2 Block Diagram

5.2 List of Pins to be Used

Table 5-1 lists the pins to be used and their functions.

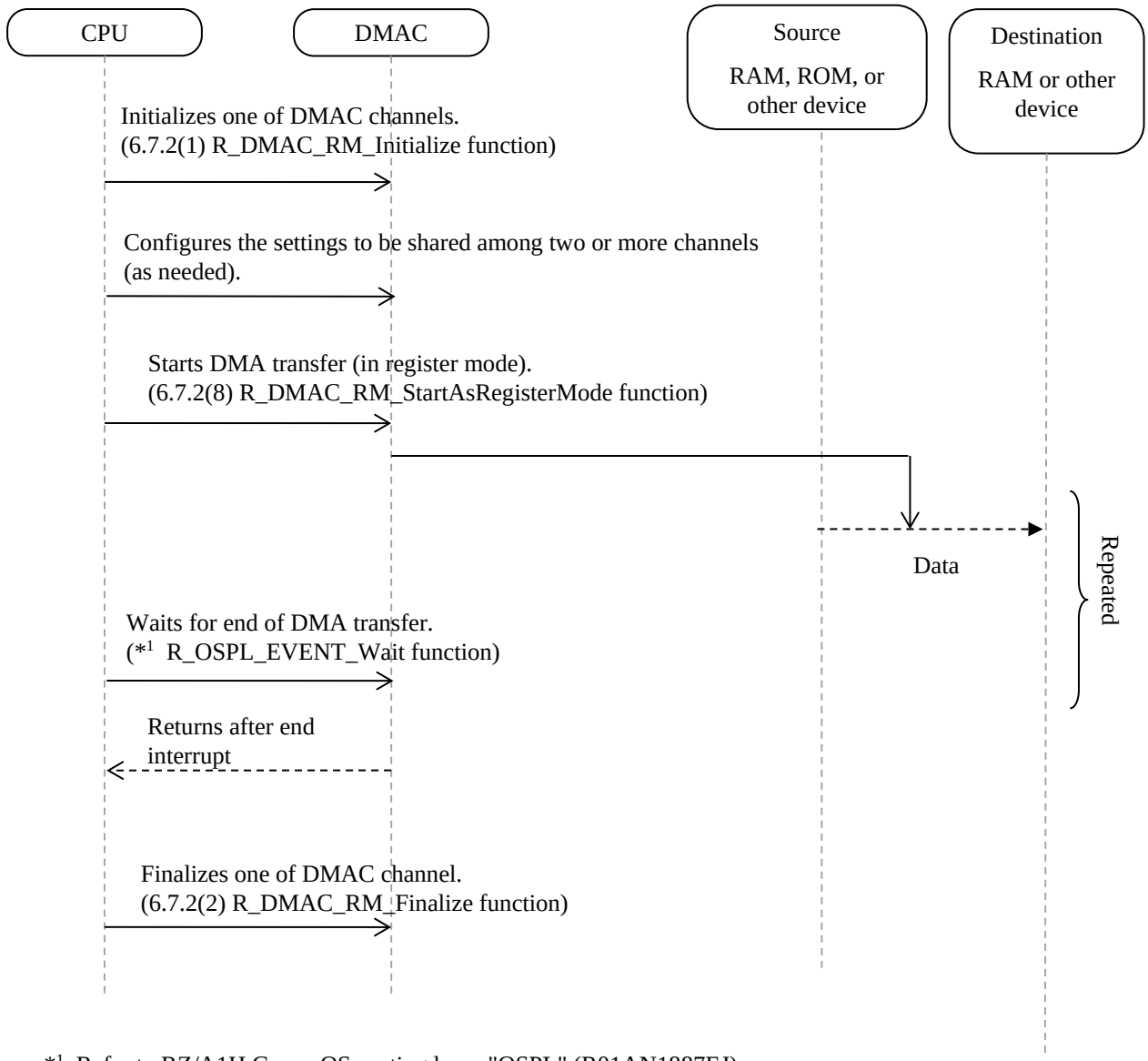
Table 5-1 Pins to be Used and their Functions

Pin name	I/O	Description
None		

6. Description of Software

6.1 Operation Outline

Figure 6-1 shows the sequence of processes.



*1 Refer to RZ/A1H Group OS porting layer "OSPL" (R01AN1887EJ)

Figure 6-1 Sequence of Processes

6.2 Interrupt

Table 6-1 shows Interrupts using by sample code.

Table 6-1 Interrupts using by sample code

Interrupt (Source ID)	Priority	Summary
DMAINT0 - DMAINT#	INTERRUPT_LEVEL_OF_INPUT_DMAC(=20) INTERRUPT_LEVEL_OF_OUTPUT_DMAC(=19)	Receives the interrupt of DMAINT#

6.3 Basic Types

Symbol	Description
char_t	8-bit character
bool_t	Logical data type. The value is true (1) or false (0).
bool8_t	8-bit logical data type.
bool16_t	16bit boolean data type. The value is true (1) or false (0).
bool32_t	32bit boolean data type. The value is true (1) or false (0).
int_t	The signed integer for this library is a 32-bit signed integer.
int8_t	8-bit signed integer (defined by standard library)
int16_t	16-bit signed integer (defined by standard library)
int32_t	32-bit signed integer (defined by standard library)
int64_t	64-bit signed integer (defined by standard library)
uint8_t	8-bit unsigned integer (defined by standard library)
uint16_t	16-bit unsigned integer (defined by standard library)
uint32_t	32-bit unsigned integer (defined by standard library)
uint64_t	64-bit unsigned integer (defined by standard library)
int_fast8_t	Fastest 8-bit minimum-width signed integer
int_fast16_t	Fastest 16-bit minimum-width signed integer
int_fast32_t	Fastest 32-bit minimum-width signed integer
uint_fast8_t	Fastest 8-bit minimum-width unsigned integer
uint_fast16_t	Fastest 16-bit minimum-width unsigned integer
uint_fast32_t	Fastest 32-bit minimum-width unsigned integer
uintptr_t	Same as pointer bit width unsigned integer as physical address
size_t	Same as pointer bit width unsigned integer as byte size
ssize_t	Same as pointer bit width signed integer
ptrdiff_t	Same as pointer bit width signed integer as difference between pointers
bit_flags_fast32_t	Same as uint_fast32_t bit flags (bit field)
bit_flags32_t	Same as uint32_t bit flags (bit field)
bit_flags16_t	Same as uint16_t bit flags (bit field)
bit_flags8_t	Same as uint8_t bit flags (bit field)
float32_t	32-bit float (Defined by standard library when "__ARM_NEON__" defined)
float64_t	64-bit float (Defined by standard library when "__ARM_NEON__" defined)
float128_t	128-bit float

6.4 Enumerations and Constants

Section	Symbol	Description
6.4.1	errnum_t	Error code
6.4.2	dmac_bit_count_t	DMA transfer size
6.4.3	dmac_transfer_mode_t	DMA transfer mode
6.4.4	dmac_ack_mode_t	DMAACK output mode
6.4.5	dmac_edge_or_level_t	Specifies whether to use the level or edge of a signal to detect a DMA request.
6.4.6	dmac_request_direction_t	Specifies whether DMAREQ should be on the source or destination side.
6.4.7	dmac_pulse_or_level_t	Specifies whether to use a pulse or level to output the DMA interrupt, DMAERR.
6.4.8	dmac_priority_mode_t	Mode for controlling the inter-channel transfer priority
6.4.9	dmac_interrupt_line_t	The interrupt line raised from DMAC
6.4.10	dmac_interrupt_lines_t	Bit flags of dmac_interrupt_line_t type
6.4.11	dmac_lm_header_t	Descriptor header of link mode
-	r_ospl_axi_cache_attribute_t	Cache attribute of AXI bus to L2 cache *1
-	r_ospl_axi_protection_t	Protection attribute of AXI bus to L2 cache *1

6.4.1 Error Codes

Symbol	Value	Description
0	0	No error is detected.
E_OTHERS	1	Others error
E_FEW_ARRAY	2	Error of few fixed length array
E_FEW_MEMORY	3	Few heap memory area
E_FIFO_OVER	4	Failed to enqueue
E_NOT_FOUND_SYMBOL	5	Not defined the symbol
E_NO_NEXT	6	There is not next element of list
E_ACCESS_DENIED	7	Error of denied read or write
E_NOT_IMPLEMENT_YET	9	Not implemented yet
E_ERRNO	0x0E(=14)	Refer to "errno"
E_LIMITATION	0x0F(=15)	Temporary limitation
E_STATE	0x10(=16)	Cannot do at this state
E_NOT_THREAD	0x11(=17)	Not a thread, Cannot call from interrupt context.
E_PATH_NOT_FOUND	0x12(=18)	Not found file or folder
E_BAD_COMMAND_ID	0x16(=22)	Out of number of command ID
E_TIME_OUT	0x17(=23)	Time out
E_STACK_OVERFLOW	0x1C(=28)	Stack overflow
E_NO_DEBUG_TLS	0x1D(=29)	Not set debug work area.
E_EXIT_TEST	0x1E(=30)	Request of exit from the test

6.4.2 dmac_bit_count_t

DMA transfer size.

*1 RZ/A1H Group OS porting layer "OSPL" Sample Program (R01AN1887EJ)

Symbol	Value	Description
DMAC_8_BIT	0	8 bits (initial value)
DMAC_16_BIT	1	16 bits
DMAC_32_BIT	2	32 bits
DMAC_64_BIT	3	64 bits
DMAC_128_BIT	4	128 bits
DMAC_256_BIT	5	256 bits
DMAC_512_BIT	6	512 bits
DMAC_1024_BIT	7	1024 bits

6.4.3 dmac_transfer_mode_t

DMA transfer mode.

Symbol	Value	Description
DMAC_SINGLE_TRANSFER_0	0	Single transfer mode (initial value)
DMAC_BLOCK_TRANSFER_1	1	Block transfer mode

6.4.4 dmac_ack_mode_t

DMAACK output mode.

Symbol	Value	Description
DMAC_ACK_MODE_RESET_VALUE	0	(initial value)
DMAC_ACK_LEVEL_MODE_001	1	Level mode
DMAC_ACK_BUS_CYCLE_MODE_010	2	Bus cycle mode
DMAC_ACK_NOT_OUTPUT_100	4	Does not output DMAACK.

6.4.5 dmac_edge_or_level_t

This specifies whether to use the level or edge of a signal to detect a DMA request.

Symbol	Value	Description
DMAC_EDGE_0	0	Uses the edge for detection. (initial value)
DMAC_LEVEL_1	1	Uses the level for detection.

6.4.6 dmac_request_direction_t

This specifies whether DMAREQ should be on the source or destination side.

Symbol	Value	Description
DMAC_DMAREQ_SOURCE_0	0	Source side. DMAACK is active at the time of read. (initial value)
DMAC_DMAREQ_DESTINATION_1	1	Destination side. DMAACK is active at the time of write.

6.4.7 dmac_pulse_or_level_t

This specifies whether to use a pulse or level to output the DMA interrupt and DMAERR.

Symbol	Value	Description
DMAC_PULSE	0	Uses the pulse to output the interrupt. (initial value)
DMAC_LEVEL	1	Uses the level to output the interrupt.

6.4.8 dmac_priority_mode_t

This is the mode for controlling the inter-channel transfer priority.

Symbol	Value	Description
DMAC_PRIORITY_FIXED	0	Fixed-priority mode. (initial value)
DMAC_PRIORITY_ROUNDROBIN	1	Round-robin mode

Note: Refer to Section 9.7.2, Priority Control for DMA Channels, of the RZ/A1H Group User's Manual: Hardware.

6.4.9 dmac_interrupt_line_t

The interrupt line raised from DMAC.

Symbol	Value	Description
DMAC_INTERRUPT_LINE_TRANSFER_END	0x00000020	End of DMA transaction
DMAC_INTERRUPT_LINE_ERROR	0x00000010	Received ERROR response in DMA transfer

6.4.10 dmac_interrupt_lines_t

Bit flags of dmac_interrupt_line_t type.

6.4.11 dmac_lm_header_t

Descriptor header of link mode.

Symbol	Value	Description
F_DMAC_LM_DESCRIPTOR_INTERRUPT_MASK	0x00000008	Usually set to 0. 0= interrupt is signaled, when "F_DMAC_LM_LINK_VALID (LV) = 0" 1= interrupt is not signaled "DEM" bit in "CHCFG_n" register in the descriptor is set whether interrupt of finished DMA transfer is signaled. Setting of interrupt must be disabled at not end of list , if DMAC was stopped or reset in interrupt callback function.
F_DMAC_LM_WRITE_BACK_DISABLE	0x00000004	Usually set to 1, if descriptor made loop. Otherwise, set to 0. 0= sets "F_DMAC_LM_LINK_VALID (LV) = 0" at the end of DMA transfer 1= does not set "F_DMAC_LM_LINK_VALID (LV) = 0". You know whether the descriptor became end of DMA transfer by reading value of "LV", if this was set to 0.

F_DMAC_LM_LINK_END	0x00000002	Set 0, if the descriptor was not end of list. Otherwise, set to 1. 0= enables "next_descriptor_link_address" for using next descriptor. 1= finishes DMA transfer at current descriptor.
F_DMAC_LM_LINK_VALID	0x00000001	Usually set to 1. 1= begins DMA transfer as descriptor 0= does not begin DMA transfer "F_DMAC_LM_LINK_VALID (LV) = 0" is done at the end of DMA transfer, if "F_DMAC_LM_WRITE_BACK_DISABLE (WBD) = 0".

6.5 Structures and Unions

Table 6-2 Structures and Unions

Section	Symbol	Outline
6.5.1	dmac_register_mode_config_t	Parameters for the "R_DMAC_RM_StartAsRegisterMode".
6.5.2	dmac_shared_config_t	Parameters for the "R_DMAC_RM_SetSharedConfiguration".
6.5.3	dmac_rm_config_t	Setting of register mode
6.5.4	dmac_lm_descriptor_t	Descriptor of link mode
6.5.5	dmac_lm_descriptor_config_t	Parameter of "R_DMAC_RM_StartAsLinkMode" function
-	r_ospl_async_t	Setting of notifications *2
-	dmac_rm_async_status_t	Same type as r_ospl_async_status_t *2

6.5.1 dmac_register_mode_config_t

Outline	Parameters for the R_DMAC_RM_StartAsRegisterMode.	
Header	r_dmac_rm.h	
Description		
Member variable	bit_flags32_t flags	Flagged structure parameters. Refer to Section 6.8.2. F_DMAC_NEXT0_SOURCE_PHYSICAL_ADDRESS F_DMAC_NEXT0_DESTINATION_PHYSICAL_ADDRESS F_DMAC_NEXT0_TRANSFER_BYTE F_DMAC_NEXT1_SOURCE_PHYSICAL_ADDRESS F_DMAC_NEXT1_DESTINATION_PHYSICAL_ADDRESS F_DMAC_NEXT1_TRANSFER_BYTE F_DMAC_IS_SOURCE_ADDRESS_FIXED F_DMAC_SOURCE_BIT_COUNT F_DMAC_IS_DESTINATION_ADDRESS_FIXED F_DMAC_DESTINATION_BIT_COUNT F_DMAC_IS_RELOAD_NEXT_REGISTER_SET F_DMAC_IS_AUTOMATIC_SWITCH_NEXT_NUM F_DMAC_NEXT_REGISTER_SET_NUMBER F_DMAC_IS_SWEEP_BUFFER F_DMAC_IS_DISABLE_NEXT_END_INTERRUPT F_DMAC_TRANSFER_MODE_TM F_DMAC_IS_DMA_ACK_OUTPUT_MODE_AM F_DMAC_REQUEST_RESOURCE_MID F_DMAC_REQUEST_RESOURCE_RID F_DMAC_REQUEST_EDGE_OR_LEVEL_LVL F_DMAC_IS_REQUEST_RAISING_OR_HIGH_HIEN F_DMAC_IS_REQUEST_FALLING_OR_LOW_LOEN F_DMAC_REQUEST_DIRECTION_REQD

*2 RZ/A1H Group OS porting layer "OSPL" Sample Program (R01AN1887EJ)

	F_DMAC_INTERVAL_COUNT F_DMAC_SOURCE_AXI_CACHE_ATTRIBUTE F_DMAC_SOURCE_AXI_PROTECTION F_DMAC_DESTINATION_AXI_CACHE_ATTRIBUTE F_DMAC_DESTINATION_AXI_PROTECTION
uintptr_t next0_source_physical_address	Value to be set in the N0SA_n register. Source address (Next0) for the next DMA transfer. Required.
uintptr_t next0_destination_physical_address	Value to be set in the N0DA_n register. Destination address (Next0) for the next DMA transfer. Required.
size_t next0_transfer_byte	Value to be set in the N0TB_n register. Total transfer byte count (Next0) for the next DMA transfer. Required.
uintptr_t next1_source_physical_address	Value to be set in the N1SA_n register. Source address (Next1) for the next DMA transfer.
uintptr_t next1_destination_physical_address	Value to be set in the N1DA_n register. Destination address (Next1) for the next DMA transfer.
size_t next1_transfer_byte	Value to be set in the N1TB_n register. Total transfer byte count (Next1) for the next DMA transfer.
bool8_t is_source_address_fixed	Value to be set in the SAD bit of the CHCFG_n register. If this variable is false, the DMA transfer source address is incremented by 1. By default, the DMA transfer destination address is incremented by 1.
dmac_bit_count_t source_bit_count	Value to be set in the SDS bit of the CHCFG_n register. DMA transfer size at the transfer source. Refer to Section 6.4.2. The default is DMAC_8_BIT.
bool8_t is_destination_address_fixed	Value to be set in the DAD bit of the CHCFG_n register. If this variable is false, the DMA transfer destination address is incremented by 1. By default, the DMA transfer destination address is incremented by 1.
dmac_bit_count_t destination_bit_count	Value to be set in the DDS bit of the CHCFG_n register. DMA transfer size at the transfer destination. Refer to Section 6.4.2. The default is DMAC_8_BIT.
bool8_t is_reload_next_register_set	Value to be set in the REN bit of the CHCFG_n register. This variable specifies whether or not to perform DMA transfer via the Next register set which is selected with next_register_set_number (the RSEL bit) after completion of the DMA transaction. The default is false.

bool8_t is_automatic_switch_next_num	Value to be set in the RSW bit of the CHCFG_n register. This variable specifies whether or not to reverse next_register_set_number (the RSEL bit) automatically after completion of the DMA transaction. The default is false.
int8_t next_register_set_number	Value to be set in the RSEL bit of the CHCFG_n register. 0 (Next0) or 1 (Next1). The default is 0.
bool8_t is_sweep_buffer	Value to be set in the SBE bit of the CHCFG_n register. This variable specifies whether or not to retrieve data from the buffer to stop the DMA transaction if enable is cleared to 0 during the DMA transaction. The default is false.
bool8_t is_disable_next_end_interrupt	Value to be set in the DEM bit of the CHCFG_n register. This variable specifies whether or not to mask the next DMA interrupt. The default is false.
dmac_transfer_mode_t transfer_mode_TM	Value to be set in the TM bit of the CHCFG_n register. DMA transfer mode. Refer to Section 6.4.3. If this variable and "request_resource_MID" variable were omitted, default value is "DMAC_BLOCK_TRANSFER_1" because of auto request mode for memory to memory. If "request_resource_MID" variable was not omitted, "transfer_mode_TM" variable cannot be omitted.
uint8_t is_DMA_ack_output_mode_AM	Value to be set in the AM bit of the CHCFG_n register. DMAACK output mode. Refer to Section 6.4.4. If this variable and "request_resource_MID" variable were omitted, the default value is "DMAC_ACK_NOT_OUTPUT_100" because of auto request mode for memory to memory. If "request_resource_MID" variable was not omitted, "transfer_mode_AM" variable cannot be omitted.
int_t request_resource_MID	Value to be set in the MID bit of the DMARSn register. Device which generates an interrupt to start DMA transfer. Refer to Section 7.5, Interrupt IDs, of the hardware manual. By default, no interaction with the device takes place, therefore DMAC transfers memory to memory as auto request mode
int_t request_resource_RID	Value to be set in the RID bit of the DMARSn register. Device which generates an interrupt to start DMA transfer. Refer to Section 7.5, Interrupt IDs, of the hardware manual. This variable is required if request_resource_MID is specified.
dmac_edge_or_level_t request_edge_or_level_LVL	Value to be set in the LVL bit of the CHCFG_n register. This variable specifies the interrupt signal waveform. Refer to Section 7.5, Interrupt IDs, of the hardware manual. This variable is required if request_resource_MID is specified.
bool8_t is_request_raising_or_high_HIEN	Value to be set in the HIEN bit of the CHCFG_n register. This variable specifies the interrupt signal waveform. Refer to Section 7.5, Interrupt IDs, of the hardware manual. This variable is required if request_resource_MID is specified.

bool8_t is_request_falling_or_low_LOEN	Value to be set in the LOEN bit of the CHCFG_n register. This variable specifies the interrupt signal waveform. Refer to Section 7.5, Interrupt IDs, of the hardware manual. This variable is required if request_resource_MID is specified.
dmac_request_direction_t request_direction_REQD	Value to be set in the REQD bit of the CHCFG_n register. This variable specifies the input or output side of the device which generates an interrupt. Refer to Section 7.5, Interrupt IDs, of the hardware manual. This variable is required if request_resource_MID is specified.
int_fast32_t interval_count	Value of setting to "ITVL" bit in "CHITVL_n" register of DMAC. Interval count. Number of counting from finish of reading or writing to next reading or writing. Unit of count is Bϕ clock. Default is 0.
r_ospl_axi_cache_attribute_t source_AXI_cache_attribute	Value of setting to "SCA" bit in "CHEXT_n" register of DMAC. Cache attribute of AXI bus, when DMAC reads via L2 cache. Default is R_OSPL_AXI_CACHE_ZERO that is for internal bus of RZ/A1H.
r_ospl_axi_protection_t source_AXI_protection	Value of setting to "SPR" bit in "CHEXT_n" register of DMAC. Protection attribute of AXI bus, when DMAC reads via L2 cache. Default is R_OSPL_AXI_PROTECTION_ZERO that is for internal bus of RZ/A1H.
r_ospl_axi_cache_attribute_t destination_AXI_cache_attribute	Value of setting to "DCA" bit in "CHEXT_n" register of DMAC. Cache attribute of AXI bus, when DMAC writes via L2 cache. Default is R_OSPL_AXI_CACHE_ZERO that is for internal bus of RZ/A1H.
r_ospl_axi_protection_t destination_AXI_protection	Value of setting to "DPR" bit in "CHEXT_n" register of DMAC. Protection attribute of AXI bus, when DMAC writes via L2 cache. Default is R_OSPL_AXI_PROTECTION_ZERO that is for internal bus of RZ/A1H.

6.5.2 dmac_shared_config_t

Outline: Parameters for the R_DMAC_RM_SetSharedConfiguration.
Header: r_dmac_rm.h
Description: Refer to the DCTRL_0_7 register description in the hardware manual.
Member variable:

bit_flags_t flags	Flagged structure parameters. Refer to Section 6.8.2. F_DMAC_LINK_WRITE_BACK_CACHE, F_DMAC_LINK_WRITE_BACK_PROTECTION, F_DMAC_LINK_DESCRIPTOR_CACHE, F_DMAC_LINK_DESCRIPTOR_PROTECTION, F_DMAC_OUTPUT_INTERRUPT_PULSE_OR_LEVEL, F_DMAC_PRIORITY_MODE
r_ospl_axi_cache_attribute_t link_write_back_cache	Value to be output to AWCACHE[3:0] for descriptor writeback in link mode. The initial value is R_OSPL_AXI_CACHE_ZERO. By omitted, no changes are made.
r_ospl_axi_protection_t link_write_back_protection	Value to be output to AWPROT[2:0] for descriptor writeback in link mode. The initial value is R_OSPL_AXI_PROTECTION_ZERO. By omitted, no changes are made.

r_ospl_axi_cache_attribute_t link_descriptor_cache	Value to be output to ARCACHE[3:0] for descriptor load in link mode. The initial value is R_OSPL_AXI_CACHE_ZERO. By omitted, no changes are made.
r_ospl_axi_protection_t link_descriptor_protection	Value to be output to ARPROT[2:0] for descriptor writeback in link mode. The initial value is R_OSPL_AXI_PROTECTION_ZERO. By omitted, no changes are made.
dmac_pulse_or_level_t output_interrupt_pulse_or_level	This variable specifies whether to use a pulse or level to output the DMA interrupt, DMAERR. Refer to Section 6.4.7. The initial value is DMAC_PULSE. By omitted, no changes are made.
dmac_priority_mode_t priority_mode	Mode for controlling the inter-channel transfer priority. Refer to Section 6.4.8. The initial value is DMAC_PRIORITY_FIXED. By omitted, no changes are made.

6.5.3 dmac_rm_config_t

Outline: Setting of register mode
Header: r_dmac_rm.h
Description: This is passed to "R_DMAC_RM_InitializeEx" function.
Member variable:

bit_flags32_t flags	Flagged structure parameters. Refer to Section 6.8.2. F_DMAC_CHANNEL_NUM
int_fast32_t channel_num	Channel number of DMAC. This is set to "channel_num" argument of "R_DMAC_RM_InitializeEx" function. This is set to unused channel number, if "channel_num" argument was "R_OSPL_UNLOCKED_CHANNEL".

6.5.4 dmac_lm_descriptor_t

Outline: Descriptor of link mode
Header: r_dmac_lm.h
Description: All member variables are set by "R_DMAC_LM_DESCRIPTOR_Initialize".
Member variable:

bit_flags32_t header	Attributes of descriptor. Bitwise OR of values as "dmac_lm_header_t"
uintptr_t source_physical_address	Value which will be set to "N0SA_n" register. Source address of DMA transfer.
uintptr_t destination_physical_address	Value which will be set to "N0DA_n" register. Destination address of DMA transfer.
uint32_t transfer_byte	Value which will be set to "N0TB_n" register. Total byte count of DMA transfer.
uint32_t CHCFG_n	Value which will be set to "CHCFG_n" register.
uint32_t CHITVL_n	Value which will be set to "CHITVL_n" register.
uint32_t CHEXT_n	Value which will be set to "CHEXT_n" register.
uintptr_t next_descriptor_link_address	Address of next descriptor

6.5.5 dmac_lm_descriptor_config_t

Outline: Parameter of "R_DMAC_RM_StartAsLinkMode" function
Header: r_dmac_lm.h
Description:

Member variable

bit_flags_fast32_t flags	Flagged structure parameters. Refer to Section 6.8.2. F_DMAC_AXI_CACHE_ATTRIBUTE, F_DMAC_AXI_PROTECTION
r_ospl_axi_cache_attribute_t AXI_cache_attribute	Cache attribute of AXI bus, when DMAC reads descriptor via L2 cache.
r_ospl_axi_protection_t AXI_protection	Protection attribute of AXI bus, when DMAC reads descriptor via L2 cache.

6.6 List of Variables

Table 6-3 shows the static variables. Table 6-4 shows the const variables.

Table 6-3 static Variables

Type	Variable Name	Contents	Function Used
r_dmac_rm_channel_t	gs_dmac_rm_channel	DMAC	Some functions

Table 6-4 const Variables

Type	Variable Name	Contents	Function Used
None			

6.7 Functions

6.7.1 List

Section	Description
(1)	List of functions which use the DMAC
(2)	List of DMAC-related functions available before initialization
(3)	List of driver porting layer functions

(1) List of functions which use the DMAC

Section	Function Name	Description
6.7.2(1)	R_DMAC_RM_Initialize	Initializes a DMAC.
6.7.2(2)	R_DMAC_RM_Finalize	Finalizes a DMAC
6.7.2(3)	R_DMAC_RM_InitializeEx	Initializes a DMAC with option.
6.7.2(4)	R_DMAC_RM_SetSharedConfiguration	Configures the settings to be shared among channels.
6.7.2(5)	R_DMAC_RM_LockChannel	Locks a channel of DMAC without initializing
6.7.2(6)	R_DMAC_RM_UnlockChannel	Unlocks a channel of DMAC without finalizing
6.7.2(7)	R_DMAC_RM_TransferAsRegisterMode	Does DMA transfer in register mode (synchronous transfer)
6.7.2(8)	R_DMAC_RM_StartAsRegisterMode	Starts DMA transfer in register mode (asynchronous transfer)
6.7.2(9)	R_DMAC_RM_StopAsRegisterMode	Stops DMA transfer which was started asynchronously in register mode
6.7.2(10)	R_DMAC_RM_TransferAsLinkMode	Does DMA transfer in link mode (synchronous transfer)
6.7.2(11)	R_DMAC_RM_StartAsLinkMode	Starts DMA transfer in link mode (asynchronous transfer)
6.7.2(12)	R_DMAC_RM_StopAsLinkMode	Stops DMA transfer which was started asynchronously in link mode
6.7.2(13)	R_DMAC_RM_OnInterrupting	Receives interrupt
6.7.2(14)	R_DMAC_RM_OnInterrupted	Handling interrupt
6.7.2(15)	R_DMAC_RM_GetAsyncStatus	Gets pointer to structure specified status of interrupt and asynchronous operation

(2) List of DMAC-related functions available before initialization

Section	Function Name	Description
6.7.3(1)	R_DMAC_RM_STATIC_GetID_FromInterruptID	Obtains a DMAC channel number from an interrupt number.
6.7.3(2)	R_DMAC_RM_STATIC_GetInterruptID	Obtains an interrupt number from a DMAC channel number.
6.7.3(3)	R_DMAC_RM_STATIC_GetDMA_BitCountSymbol	Obtains a symbol for DMA transfer size.
6.7.3(4)	R_DMAC_LM_DESCRIPTOR_Initialize	Makes a descriptor of link mode

(3) List of driver porting layer functions

Section	Function Name	Description
6.7.4(1)	R_DMAC_RM_SetDefaultAsync	Sets default value of r_ospl_async_t type structure.
6.7.4(2)	R_DMAC_RM_OnInitialize	Initializes the driver porting layer part.
6.7.4(3)	R_DMAC_RM_OnFinalize	Performs termination processing for the driver porting layer part.
6.7.4(4)	R_DMAC_RM_SetInterruptCallbackCaller	Sets interrupt callback function caller to driver porting layer part
6.7.4(5)	R_DMAC_RM_OnEnableInterrupt	Enables interrupts
6.7.4(6)	R_DMAC_RM_OnDisableInterrupt	Disables interrupts
6.7.4(7)	R_DMAC_RM_OnInterruptDefault	Default interrupt callback function

6.7.2 Functions Which Use the DMAC

(1) R_DMAC_RM_Initialize

Outline	Initializes a DMAC.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_Initialize(int_fast32_t dmac_channel);	
Description	Initializes the internal variables. Clears the status register (SWRST bit).	
Arguments	int_fast32_t dmac_channel	DMAC channel number
Return value	Error code. If there is no error, the return value is 0	

(2) R_DMAC_RM_Finalize

Outline	Finalizes a DMAC.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_Finalize(int_fast32_t dmac_channel, errnum_t e);	
Description	Finalizes the internal variables. Stop DMA transfer, if DMA transfer was running.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	errnum_t e	Error code which is had raised. 0=No error
Return value	Error code or e. 0 = successful and "e = 0".	

(3) R_DMAC_RM_InitializeEx

Outline	Initializes a DMAC with option.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_InitializeEx(int_fast32_t dmac_channel, dmac_rm_config_t* in_out_Config);	
Description	Initializes the internal variables. Clears the status register (SWRST bit). This function can be set with "dmac_rm_config_t" type option.	
Arguments	int_fast32_t dmac_channel	DMAC channel number. Unused channel number is set in "in_out_Config->channel_num", if "R_OSPL_UNLOCKED_CHANNEL." was specified.
	dmac_rm_config_t* in_out_Config	NULL or "dmac_rm_config_t". See 6.5.3
Return value	Error code. If there is no error, the return value is 0	

(4) R_DMAC_RM_SetSharedConfiguration

Outline	Configures the settings to be shared among channels.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_SetSharedConfiguration(int_fast32_t dmac_channel, dmac_shared_config_t* config);	
Description		
Arguments	int_fast32_t dmac_channel	DMAC channel number If 0 to 7 are specified, the settings to be shared among channels 0 to 7 are configured. If 8 to 15 are specified, the settings to be shared among channels 8 to 15 are configured.
	dmac_shared_config_t* config	Setting. Refer to Section 6.5.2.
Return value	Error code. If there is no error, the return value is 0	

(5) R_DMAC_RM_LockChannel

Outline	Locks a channel of DMAC without initializing	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_LockChannel(int_fast32_t const dmac_channel, int_fast32_t* out_ChannelNum);	
Description	This function removes specified channel from unused channel list and another driver can use the channel and does not conflict with the channel, if channel was locked without initializing.	
Arguments	int_fast32_t const dmac_channel	Channel number which will be locked or "R_OSPL_UNLOCKED_CHANNEL"
	int_fast32_t* out_ChannelNum	Channel number which was locked
Return value	Error code. If there is no error, the return value is 0	

(6) R_DMAC_RM_UnlockChannel

Outline	Unlocks a channel of DMAC without finalizing	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_UnlockChannel(int_fast32_t const dmac_channel, errnum_t e);	
Description		
Arguments	int_fast32_t const dmac_channel	Channel number which will be unlocked
	errnum_t e	Error code which is had raised. 0=No error
Return value	Error code or e. 0 = successful and "e = 0".	

(7) R_DMAC_RM_TransferAsRegisterMode

Outline	Does DMA transfer in register mode (synchronous transfer)	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_TransferAsRegisterMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* config);	
Description	This is synchronous function which does not return until end of transfer. This writes 1 to "STG" bit in "CHCTRL_n" register as auto request mode, if "config->request_resource_MID" was omitted.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_register_mode_config_t* config	Setting. Refer to Section 6.5.1.
Return value	Error code. If there is no error, the return value is 0	

(8) R_DMAC_RM_StartAsRegisterMode

Outline	Starts DMA transfer in register mode (asynchronous transfer)	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_StartAsRegisterMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* config, r_ospl_async_t* Async);	
Description	This is asynchronous function which returns before end of transfer. The other specification is same as "R_DMAC_RM_StartAsRegisterMode". Regarding "Async" argument, See "R_DRIVER_TransferAsync" in RZ/A1H group OS porting layer "OSPL" R01AN1887EJ.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_register_mode_config_t* config	Setting. Refer to Section 6.5.1.
	r_ospl_async_t* Async	Setting of notification

Return value	Error code. If there is no error, the return value is 0

(9) R_DMAC_RM_StopAsRegisterMode

Outline	Stops DMA transfer which was started asynchronously in register mode	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_StopAsRegisterMode(int_fast32_t dmac_channel);	
Description	Event specified with "Async" argument of "R_DMAC_RM_StartAsRegisterMode" function will not be signaled, if stopped.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
Return value	Error code. If there is no error, the return value is 0.	

(10) R_DMAC_RM_TransferAsLinkMode

Outline	Does DMA transfer in link mode (synchronous transfer)	
Header	r_dmac_lm.h	
Declaration	errnum_t R_DMAC_RM_TransferAsLinkMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* a_config, uintptr_t first_descriptor_physical_address, dmac_lm_descriptor_config_t* descriptor_config);	
Description	DMA transfers are started sequentially which specified with descriptor chained by list structure (link mode). This is synchronous function which does not return until end of all transfers. This writes 1 to "STG" bit in "CHCTRL_n" register as auto request mode, if "config->request_resource_MID" was omitted. Writing to 2nd or more descriptor is not done because of not necessary.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_register_mode_config_t* a_config	Settings of peripheral function which signals interrupt for triggering with DMA transfer. Member variables except for following them are ignored. ● flags ● request_resource_MID ● request_resource_RID This argument can be specified with one of reading descriptor.
	uintptr_t first_descriptor_physical_address	Physical address of first reading descriptor as "dmac_lm_descriptor_t*" type.
	dmac_lm_descriptor_config_t* descriptor_config	Settings of descriptor. See 6.5.5.
Return value	Error code. If there is no error, the return value is 0.	

(11) R_DMAC_RM_StartAsLinkMode

Outline	Starts DMA transfer in link mode (asynchronous transfer)
Header	r_dmac_lm.h
Declaration	errnum_t R_DMAC_RM_StartAsLinkMode(int_fast32_t dmac_channel, dmac_register_mode_config_t* a_config, uintptr_t first_descriptor_physical_address, dmac_lm_descriptor_config_t* descriptor_config, r_ospl_async_t* Async);
Description	This is asynchronous function which returns before end of transfer. The other specification is same as "R_DMAC_RM_StartAsRegisterMode". Regarding "Async" argument, See "R_DRIVER_TransferAsync" in RZ/A1H group OS porting layer "OSPL" R01AN1887EJ.

Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_register_mode_config_t* a_config	Settings of peripheral function which signals interrupt for triggering with DMA transfer. Member variables except for following them are ignored. ● flags ● request_resource_MID ● request_resource_RID This argument can be specified with one of reading descriptor.
	uintptr_t first_descriptor_physical_address	Physical address of first reading descriptor as "dmac_lm_descriptor_t*" type.
	dmac_lm_descriptor_config_t* descriptor_config	Settings of descriptor. See 6.5.5.
	r_ospl_async_t* Async	Setting of notification
Return value	Error code. If there is no error, the return value is 0.	

(12) R_DMAC_RM_StopAsLinkMode

Outline	Stops DMA transfer which was started asynchronously in link mode	
Header	r_dmac_lm.h	
Declaration	errnum_t R_DMAC_RM_StopAsLinkMode(int_fast32_t dmac_channel);	
Description	Event specified with "Async" argument of "R_DMAC_RM_StartAsLinkMode" function will not be signaled, if stopped.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
Return value	Error code. If there is no error, the return value is 0.	

(13) R_DMAC_RM_OnInterrupting

Outline	Receives interrupt	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_OnInterrupting(const r_ospl_interrupt_t* InterruptSource);	
Description	This function sets interrupt notification from interrupt status register to "dmac_rm_async_status_t::InterruptFlags" variable and clears the interrupt. See "R_DRIVER_OnInterrupting" function in RZ/A1H group OS porting layer "OSPL" R01AN1887EJ.	
Arguments	r_ospl_interrupt_t* InterruptSource	Source of interrupt
Return value	Error code. If there is no error, the return value is 0.	

(14) R_DMAC_RM_OnInterrupted

Outline	Handling interrupt	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_OnInterrupted(int_fast32_t const dmac_channel);	
Description	This function clears bit in "dmac_rm_async_status_t::InterruptFlags" variable which was set by "R_DMAC_RM_OnInterrupting", therefore this function handles interrupt. See "R_DRIVER_OnInterrupted" function in RZ/A1H group OS porting layer "OSPL" R01AN1887EJ.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
Return value	Error code. If there is no error, the return value is 0.	

(15) R_DMAC_RM_GetAsyncStatus

Outline	Gets pointer to structure specified status of interrupt and asynchronous operation	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_GetAsyncStatus(int_fast32_t dmac_channel, const dmac_rm_async_status_t** out_Status);	
Description	It is necessary to write "const" qualifier with pointer variable specifying with "out_Status" argument.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_rm_async_status_t** out_Status	Output: Pointer to structure specified status of interrupt and asynchronous operation
Return value	Error code. If there is no error, the return value is 0.	

6.7.3 DMAC-related functions available before initialization

(1) R_DMAC_RM_STATIC_GetID_FromInterruptID

Outline	Obtains a DMAC channel number from an interrupt number.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_STATIC_GetID_FromInterruptID(bsp_int_src_t interrupt_id, int_fast32_t* out_dmac_channel);	
Description		
Arguments	bsp_int_src_t interrupt_id	Interrupt number
	int_fast32_t* out_dmac_channel	Output: DMAC channel number
Return value	Error code. If there is no error, the return value is 0.	

(2) R_DMAC_RM_STATIC_GetInterruptID

Outline	Obtains an interrupt number from a DMAC channel number.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_STATIC_GetInterruptID(int_fast32_t dmac_channel, bsp_int_src_t* out_interrupt_id);	
Description	This function cannot be used to obtain DMA interrupt status.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	bsp_int_src_t* out_interrupt_id	Output: Interrupt number
Return value	Error code. If there is no error, the return value is 0.	

(3) R_DMAC_RM_STATIC_GetDMA_BitCountSymbol

Outline	Obtains a symbol for DMA transfer size.	
Header	r_dmac_rm.h	
Declaration	errnum_t R_DMAC_RM_STATIC_GetDMA_BitCountSymbol(int_t bit_count, dmac_bit_count_t* out_symbol);	
Description		
Arguments	int_t bit_count	DMA transfer size (in bytes)
	dmac_bit_count_t* out_symbol	Output: Symbol for the DMA transfer size
Return value	Error code. If there is no error, the return value is 0.	

(4) R_DMAC_LM_DESCRIPTOR_Initialize

Outline	Makes a descriptor of link mode
Header	r_dmac_lm.h

Declaration	errnum_t R_DMAC_LM_DESCRIPTOR_Initialize(dmac_lm_descriptor_t* self, int_fast32_t const dmac_channel, bit_flags32_t header, dmac_register_mode_config_t* config, uintptr_t next_descriptor_physical_address);
Description	Flush cache of the memory area specified with "self" argument after calling this function before starting DMA transfer as link mode by calling "R_DMAC_RM_TransferAsLinkMode" or other function.

Arguments

dmac_lm_descriptor_t* self	Virtual address of the variable as "dmac_lm_descriptor_t" type structure. Output: descriptor for link mode
int_fast32_t const dmac_channel	DMAC channel number which starts DMA transfer specified with "config" argument
bit_flags32_t header	Bitwise OR as "dmac_lm_header_t"
dmac_register_mode_config_t* config	Settings of DMA transfer. See 6.5.1. But member variable which name is started from "F_DMAC_NEXT1_*" cannot be specified.
uintptr_t next_descriptor_physical_address	Physical address of the descriptor which is read after the end of DMA transfer specified with "self" argument.

Return value: Error code. If there is no error, the return value is 0.

6.7.4 Driver porting layer functions

(1) R_DMAC_RM_SetDefaultAsync

Outline	Sets default value of r_ospl_async_t type structure.
Header	r_dmac_rm_pl.h
Declaration	void R_DMAC_RM_SetDefaultAsync(r_ospl_async_t* Async);
Description	This function sets each member variable to default value related to bit value of Flags member variable in r_ospl_async_t type struct is 0.

Arguments

r_ospl_async_t* Async	The setting of notifications. NULL is not permitted.

Return value: None

(2) R_DMAC_RM_OnInitialize

Outline	Initializes the driver porting layer part.
Header	r_dmac_rm_pl.h
Declaration	errnum_t R_DMAC_RM_OnInitialize(int_fast32_t dmac_channel);
Description	If necessary, supply the clock. This function is called from the R_DMAC_RM_Initialize.

Arguments

int_fast32_t dmac_channel	DMAC channel number

Return value: Error code. If there is no error, the return value is 0.

(3) R_DMAC_RM_OnFinalize

Outline	Performs termination processing for the driver porting layer part.
Header	r_dmac_rm_pl.h
Declaration	errnum_t R_DMAC_RM_OnFinalize(int_fast32_t dmac_channel, errnum_t e);
Description	If necessary, stop to supply the clock. This function is called from the R_DMAC_RM_Finalize function.

Arguments

int_fast32_t dmac_channel	DMAC channel number
errnum_t e	Errors that have occurred. No error = 0

Return value: Error code or e 0 = successful and e = 0.

(4) R_DMAC_RM_SetInterruptCallbackCaller

Outline	Sets interrupt callback function caller to driver porting layer part

Header	r_dmac_rm_pl.h	
Declaration	errnum_t R_DMAC_RM_SetInterruptCallbackCaller(int_fast32_t dmac_channel, const r_ospl_caller_t* caller);	
Description	This function is called by each starting to asynchronous operation with interrupts. Call R_OSPL_CallInterruptCallback function with "caller" argument of this function from interrupt handler. Refer to the section of R_OSPL_CallInterruptCallback function in RZ/A1H Group OS porting layer "OSPL" (R01AN1887JJ) It is not necessary to check channel number in this function.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	r_ospl_caller_t* caller	The value passing to R_OSPL_CallInterruptCallback
Return value	Error code. If there is no error, the return value is 0	

(5) R_DMAC_RM_OnEnableInterrupt

Outline	Enables interrupts	
Header	r_dmac_rm_pl.h	
Declaration	void R_DMAC_RM_OnEnableInterrupt(int_fast32_t dmac_channel, dmac_interrupt_lines_t enables);	
Description	It is not necessary to check channel number in this function.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_interrupt_lines_t enables	Bit flags set to 1 enabling interrupt line
Return value	None	

(6) R_DMAC_RM_OnDisableInterrupt

Outline	Disables interrupts	
Header	r_dmac_rm_pl.h	
Declaration	void R_DMAC_RM_OnDisableInterrupt(int_fast32_t dmac_channel, dmac_interrupt_lines_t disables);	
Description	It is not necessary to check channel number in this function.	
Arguments	int_fast32_t dmac_channel	DMAC channel number
	dmac_interrupt_lines_t disables	Bit flags set to 1 disabling interrupt line
Return value	None	

(7) R_DMAC_RM_OnInterruptDefault

Outline	Default interrupt callback function	
Header	r_dmac_rm_pl.h	
Declaration	errnum_t R_DMAC_RM_OnInterruptDefault(const r_ospl_interrupt_t* interrupt_source, const r_ospl_caller_t* caller);	
Description	This function will be called when InterruptCallback member variable in async argument of R_DMAC_RM_TransferAsync function was NULL. This function is r_ospl_callback_t type.	
Arguments	r_ospl_interrupt_t* interrupt_source	Interrupt source
	r_ospl_caller_t* caller	The value passed to R_OSPL_CallInterruptCallback
Return value	Error code. If there is no error, the return value is 0	

6.8 Supplementary Information

6.8.1 Cooperation with RZ/A1H RTX BSP

It is necessary to unified manage to lock channels, if DMAC_RM driver and RZ/A1H RTX BSP were used at the same time. DMAC_RM driver locks channel of DMAC driver (by calling "R_DMA_Alloc" API) in BSP via OSPL, therefore BSP manages them, if "IS_RZ_A1_BSP_USED = 1".

6.8.2 Flagged structure parameters

Flags member variables in the structure are used as bit flags, and if a bit is 1, the corresponding member variable is enabled according to the coding pattern. If a bit is 0, the value of the member variable is assumed to be the default value. Even if the version is upgraded so that its structure contains additional members, the old and new versions can be binary compatible.

```
FuncA_ConfigClass  config;

config.Flags  = F_FuncA_Param1 | F_FuncA_Param2;
config.Param1 = 10;
config.Param2 = 2;
FuncA( &config );
```

Because there is not Flags |= F_FuncA_Param3, config.Param3 is the default value.

7. Sample Codes

The sample codes can be downloaded from the Renesas Electronics website.

8. Documents for Reference

User' manual: Hardware

RZ/A1H Group User's Manual: Hardware

The latest version can be downloaded from the Renesas Electronics website.

R7S72100 RTK772100BC00000BR (GENMAI) User's Manual

The latest version can be downloaded from the Renesas Electronics website.

R7S72100 CPU (GENMAI) Optional Board RTK7721000B00000BR User's Manual

The latest version can be downloaded from the Renesas Electronics website.

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

The latest version can be downloaded from the ARM website.

ARM Generic Interrupt Controller Architecture Specification Architecture version 1.0

The latest version can be downloaded from the ARM website.

Technical Updates/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

ARM Software Development Tools (ARM Compiler toolchain, ARM DS-5 etc.) can be downloaded from the ARM website.

The latest version can be downloaded from the ARM website.

Website and Support

Renesas Electronics website
http://www.renesas.com

Inquiries
http://www.renesas.com/contact/

Revision History

Rev.	Date	Description
1.03	Aug. 24, 2017	Updated internal OSPL to version 1.60.
1.02	Feb. 29, 2016	Added following items: ● Link mode ● Locking channels ● Searching unused channel (R_OSPL_UNLOCKED_CHANNEL) ● Supporting L2 cache (e.g. F_DMAC_DESTINATION_AXI_CACHE_ATTRIBUTE) Updated initial settings to version 1.01. Updated internal OSPL to version 0.96. Fixed issue of state transition in "R_DMAC_RM_StopAsRegisterMode".
1.00	Jun. 20, 2014	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
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; and 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.

   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 (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident 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 Renesas Electronics 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 ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, 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 investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
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 majority-owned subsidiaries.

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

(Rev.3.0-1 November 2016)

RENESAS

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.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, P. R. China 200333
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, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, 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 India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0