Application Note

RENESAS

RSCAN – Lite Software Driver

RL78

16-bit Microcontroller

Renesas Electronics
www.renesas.com

R01AN1686ED0100, Rev. 01.00
June 2013

Table of Contents

Chapter 1 Introduction

RSCAN Family RSCAN-Lite (One Channel) Low End CAN Controller

Product RL78–RSCAN-Lite Software Driver

Usage This document provides further support information for customers using a CAN macro from the RSCAN family line.

Overview At this application note a basic functional description of the RSCAN-Lite is given. Further a complete C based software driver plus API description is provided.

The practical experience for the user is improved by application and test functions based of the API.

Chapter 2 RSCAN-Lite general feature descriptions

RS-CAN - Lite The RSCAN-Lite is a low end CAN Controller of the RSCAN family. In figure 1 the functional block diagram of the RSCAN-Lite is shown.

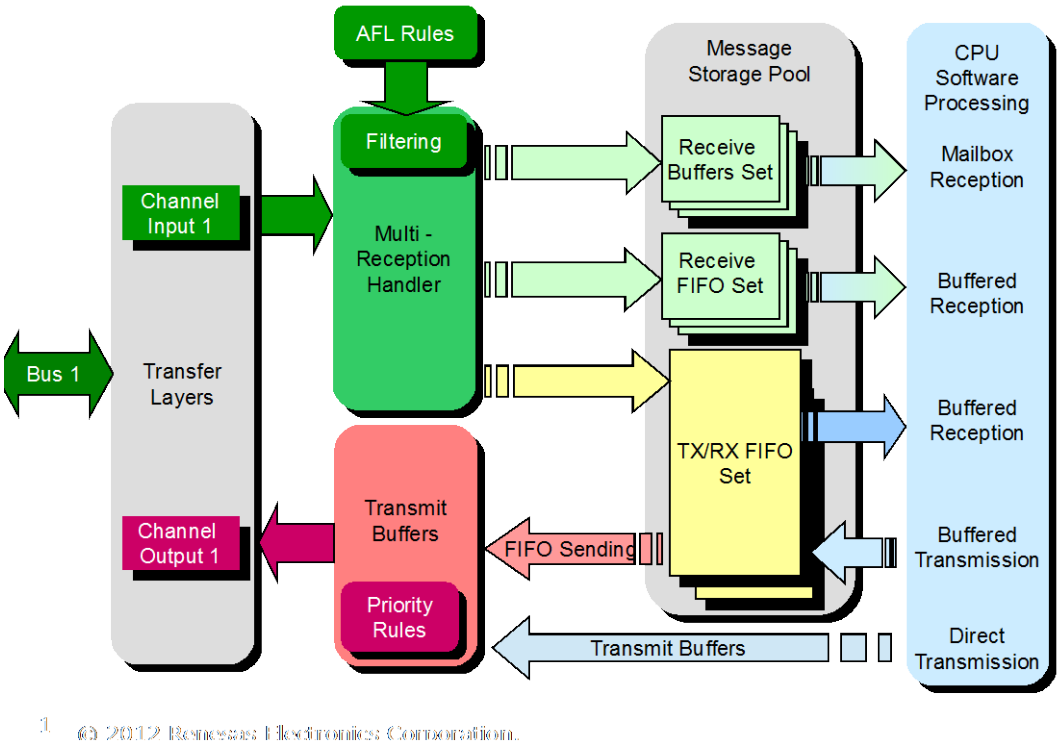

Illustration 1: Functional Block RSCAN

Inside the block, the transfer layer handles the protocol depended tasks for every channel. In case of the RL78, this is only one channel with reception and transmit path. All the incoming messages are handled by the multi-reception handler. The handler filters and maps the messages to the dedicated buffers in the message storage pool. For coordinating the outgoing messages the transmit buffer controls the output stream based on the priority rules.

Overall the functionality is split into two groups: Transmission path and reception path.

Reception Path All incoming messages to the RSCAN channel are processed by the transfer layer and multi reception handler. To filter the incoming messages the acceptance filter rules are used. The AFL rules are stored at the RAM and are accessible for the RSCAN. Further the AFL rules define the storage path for the filtered messages.

The message storing place could be divided into two classes of buffers: Normal Buffers and FIFO Buffers. The receive buffers handle the messages like a mailbox. Therefore it is required to read or save incoming messages, before a new message comes in. The second class of buffers are the FIFO's. The RSCAN has two kinds of FIFO's: Receive FIFO and TX/RX FIFO. The FIFO's could receipt the incoming message like a stream format. All incoming messages are saved at the buffer, till the maximum FIFO size is reached.

Transmission Path

All outgoing messages are handled by the transmission path. The main components at the transmission path are the transmit buffers. They coordinate the order of the outgoing messages by the priority rules. The priority order depends of the transmit buffer or Message-ID. General three types of transmission are possible: Normal Transmission, FIFO Transmission and Gateway Transmission.

For further description, please refer to the corresponding User Manual.

Chapter 3 RSCAN-Lite Software Driver

In this chapter, first an overview of the SW Driver Structure is given. After that a detailed description of the corresponding API function follows.

3.1 SW Driver Structure

File Structure The Table 1 shows all source files available and ordered.

Path or File:	Description:
.\ include \ ..	Header Path
.\ device.h	Device Specific Header and Environment Includes
.\ map_rscan.h	Macro Size, Address and Clock definition (mapping)
.\ map_rscan_irq.h	Interrupt Service Routines (mapping)
.\ map_rscan_irqbundling.h	Interrupt Handling Function
.\ ree_types.h	Renesas Defines (Type-definitions)
.\ ..	Root Path / Source Path
.\ rscan.h	Main Macro SW Header (Register Structure + SW Definitions)
.\ rscan_p.h	Driver Header (API)
.\ rscan_p.c	Driver Source
.\ rscan_s.h	Application Data (Configurations, AFL)
.\ rscan_a.h	Application Test Header
.\ rscan_a.c	Application Test Source

Table1: File Structure of API, Test and Application Source Files

Mapping There are three files used, to provide the mapping compatibility for the driver:

- device.h
- map_rscan.h
- map_rscan_irq.h

The “device.h” contains the device specific header files. This comprise the interrupt control register definitions and port names. Further, compiler dependent includes are defined.

The opportunity to configure the driver to various products and macro sizes, the “map_rscan.h” file is used. Also the memory mapping is defined in this file.

Device dependent interrupt sources and their service routines are defined in “map_rscan_irq.h”

Interrupt Handling

To manage interrupt handling of the RSCAN, an interrupt bundling function is defined. This function handles the interrupt lines and reset the interrupt request flags. The function is defined in “map_rscan_irq.h”. The function delivers a global variable with a request flag, set in the associated interrupt service routine.

Macro Register

The register mapping of the RSCAN is defined in “rscan.h”. It contains several type specifications, to depict all register from the macro.

All register groups are merged in the main type-structure “ee_rscan_common”. This type is used to define a pointer at the basic offset address. This allows the whole access of all Registers over the “ee_rscan_common_p” pointer. The register / memory mapping with pointer and address specification could be found in “map_rscan.h”.

3.2 RSCAN Lite SW Driver API

3.2.1 API predefined Data

Low Level API Functions

The following data is used for RSCAN configuration and can be found in “rscan_s.h”. It could be used for application and test software or as an example.

The schema of the data table documentation is explained at table 2.

Type Name_of_Data			
Description:	*General Description and Usage of the Data. Example Data could be found in “rscan_s.h”.*		
Name:	Type Structure / Data :	Size:	Register:
Name_of_Structure_Item_0	*Type_of_Structure_Item_0*	*Bit_size*	*Register_Name of Item 0*
...	...	...	...
Name_of_Structure_Item_n	*Type_of_Structure_Item_n*	*Bit_size*	*Register_Name of Item n*

Table2: Schema of Data Description

const struct ee_rscan_cfg_global EE_RSCAN_GCFG_xx			
Description:	*The constant data type contains the Global Configuration of the Macro.*		
Name:	Type Structure / Data :	Size:	Register:
gcfg	*ee_rscan_c_gcfg*	*32 bit*	*Global Configuration Register*
gctr	*ee_rscan_c_gctr*	*32 bit*	*Global Control Register*
rmnb	*Unsigned Integer*	*32 bit*	*Receive Buffer Number Configuration Register*
rnc [EE_RSCAN_CHANNELS]	*Unsigned Integer Array*	*1x 32 bit*	*Receive Rule Number Set Register*
rfcc [EE_RSCAN_MAXRXFIFOS]	*ee_rscan_c_rfcc Array*	*2 x 32 bit*	*Receive FIFO control Register*

Table 3: Global Configuration Data

const struct ee_rscan_cfg_channel EE_RSCAN_CHCFG_xx			
Description:	*The constant data type contains the Channel Configuration.*		
Name:	Type Structure / Data :	Size:	Register / Property:
bitrate	*Unsigned Integer*	*32 bit*	*This Value contains the Bitrate to be configured per SW.*
tq_perbit	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
syncjumpwidth	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
samplingpointpos	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
ctr	*ee_rscan_c_ctr*	*32 bit*	*Channel Control Register*
tmiec	*Unsigned Short*	*16 bit*	*Transmit Buffer Interrupt Enable Register*
thlcc	*ee_rscan_c_thlcc*	*16 bit*	*Transmit History Buffer Control Register*
cfcc	*ee_rscan_c_cfcc*	*32 bit*	*Transmit/Receive FIFO Control Register*

Table 4: Channel Configuration Data

const struct ee_rscan_a_afl EE_RSCAN_AFL_xx			
Description:	*The constant data type contains the Channel Configuration.*		
Name:	Type Structure / Data :	Size:	Register / Property:
bitrate	*Unsigned Integer*	*32 bit*	*This Value contains the Bit Rate to be configured per SW.*
tq_perbit	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
syncjumpwidth	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
samplingpointpos	*Unsigned Char*	*8 bit*	*Optional Value. Calculated by SW.*
ctr	*ee_rscan_c_ctr*	*32 bit*	*Channel Control Register*
tmiec	*Unsigned Short*	*16 bit*	*Transmit Buffer Interrupt Enable Register*
thlcc	*ee_rscan_c_thlcc*	*16 bit*	*Transmit History Buffer Control Register*
cfcc	*ee_rscan_c_cfcc*	*32 bit*	*Transmit/Receive FIFO Control Register*

Table 5: Acceptance Filter Entry

struct ee_rscan_message SendMessage_xx or ReceiptMessage_xx			
Description:	*This data type is used to define Message-Data.*		
Name:	Type Structure / Data :	Size:	Register / Property:
hdr	*ee_rscan_t_mask*	*32 bit*	*ID, IDE, RTR and THLEN (only for TX)*
flag	*ee_rscan_r_ptr*	*32 bit*	*TS, PTR and DLC*
data[EE_RSCAN_DLC_MAX]	*Unsigned Char*	*8x 8 bit*	*Data of the Message*
path	*Unsigned Char*	*8 bit*	*EE_RSCAN_PATH_MSGBOX*
			EE_RSCAN_PATH_COMFIFO
			EE_RSCAN_PATH_RXFIFO
			EE_RSCAN_PATH_ANY
pathdetail	*Unsigned Char*	*8 bit*	*Buffer Number*

Table 6: Storage Data for Messages (Transmission, Reception)

3.2.2 Low Level API Functions

Return_type Function_Name(function_variable_0, …, function_variable_n);				
Description:	*Description of the Function.*			
Direction:	Parameter:	Type:	Description:	Value:
Input:	*Variable_0*	*Variable Type*	*Variable Description 0*	*Macro Specific Value or Enumerated Value*
	...	...	...	...
	Variable_n	*Variable Type*	*Variable Description n*	*Macro Specific Value or Enumerated Value*
Output:	*Return*	*Bit (REE Type)*	*Return value gives a true or false back. It's a function error check. This mechanism is given for all functions at the API.*	*True = check result*
				False = check result

Table 7: Schema of Function Description

extern bit EE_RSCAN_Port_Enable(u08 Unit_u08, u08 Channel_u08);				
Description:	*This Function configures the Channel associated Port. The transmit and receive port are configured.*			
Direction:	Parameter:	Type:	Description:	Value:
Input:	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
Output:	*Return*	*Bit*	*The Return value gives a true or false callback. The Value returns the function fault check.*	*True*
				False

Table 8: Enable dedicated Port for Channel

extern bit EE_RSCAN_Global_Mode_change(u08 Unit_u08, u08 Global_Mode_u08);				
Description:	*This Function changes the Global Mode of the Macro.*			
Direction:	Parameter:	Type:	Description:	Value:
Input:	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Global_Mode_u08	*Unsigned Char*	*Transition to Global Mode*	*EE_RSCAN_GLOBAL_OPER*
				EE_RSCAN_GLOBAL_RST
				EE_RSCAN_GLOBAL_TST
				EE_RSCAN_GLOBAL_SLEEP
Output:	*Return*	*Bit*	*The Return value gives a true or false callback. The Value returns the function fault check.*	*True*
				False

Table 9: Change Global Mode of Macro

extern bit EE_RSCAN_Channel_Mode_change(u08 Unit_u08, u08 Channel_u08, u08 Channel_Mode_u08);				
Description:	*This Function changes the Mode of the specific Channel.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Channel_Mode_u08	*Unsigned Char*	*Transition to Channel Mode*	*EE_RSCAN_CH_OPER*
				EE_RSCAN_CH_RESET
				EE_RSCAN_CH_HALT
				EE_RSCAN_CH_KEEP
				EE_RSCAN_CH_SLEEP
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 10: Change dedicated Channel Mode

extern bit EE_RSCAN_Config(u08 Unit_u08, u08 Channel_u08, const struct ee_rscan_cfg_global Global_CFG, const struct ee_rscan_cfg_channel* Ch_CFG, struct ee_rscan_a_afl FilterEntry[]);*				
Description:	*This Function configures the hole Macro. Therefore the Global and Channel Configurations are passed by a Pointer. Further the AFL Rules are passed. After that the Global and Channel Modes are in Operation Mode.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Global_CFG	*const struct ee_rscan_cfg_global**	*Global Configuration of the RSCAN Macro.*	*Global Configuration*
				Global Control
				RMNB
				RNC
				RFCC
	Ch_CFG	*const struct ee_rscan_cfg_channel**	*Channel Configuration for specific Channel_u08.*	*Bitrate*
				TQ per Bit
				Synchronization jump width
				Sampling Point Position
				Channel Control
				TMIEC
				THLCC
				CFCC
	FilterEntry[]	*struct ee_rscan_a_afl*	*All AFL Rules are defined for the specific Channel.*	*ID*
				MASK
				PTR0
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 11: Function to configure the hole Macro in one shot

extern bit EE_RSCAN_Config_Self_Test(u08 Unit_u08, u08 Channel_u08, const struct ee_rscan_cfg_global Global_CFG, const struct ee_rscan_cfg_channel* Ch_CFG, struct ee_rscan_a_afl FilterEntry[]);*				
Description:	*This Function configures the hole Macro. Therefore the Global and Channel Configurations are passed by a Pointer. Further the AFL Rules are passed. After that the Global and Channel Modes are in Test Mode. Further the Channel is in intern or external loop back.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Global_CFG	*const struct ee_rscan_cfg_global**	*Global Configuration of the RSCAN Macro.*	*Global Configuration* *Global Control* *RMNB* *RNC* *RFCC*
	Ch_CFG	*const struct ee_rscan_cfg_channel**	*Channel Configuration for specific Channel_u08.*	*Bitrate* *TQ per Bit* *Synchronization jump width* *Sampling Point Position* *Channel Control* *TMIEC* *THLCC* *CFCC*
	FilterEntry[]	*struct ee_rscan_a_afl*	*All AFL Rules are defined for the specific Channel.*	*ID* *MASK* *PRT0*
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True* *False*

Table 12: Function to configure the whole Macro in one shot for Self-Test

*extern bit EE_RSCAN_CreateInterrupt(u08 Unit_u08, u08 Channel_u08, u08 IntNumber_u08, u08 SetIntLevel_u08, void (*FunctionVector)());*				
Description:	*This Function enables Interrupt Sources and bundles the Interrupt with a Function.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	IntNumber_u08	*Unsigned Char*	*Interrupt Source*	*EE_RSCAN_INT_GERR*
				EE_RSCAN_INT_RXF0
				EE_RSCAN_INT_RXF1
				EE_RSCAN_INT_TX
				EE_RSCAN_INT_TXA
				EE_RSCAN_INT_TXQ
				EE_RSCAN_INT_CERR
				EE_RSCAN_INT_TXHL
				EE_RSCAN_INT_RXCF
				EE_RSCAN_INT_TXCF
	SetIntLevel_u08	*Unsigned Char*	*Interrupt Priority Level*	*0 … 3*
	*(*FunctionVector)()*	*Depends on function*	*Pointer to bundled Interrupt Function*	*Function*
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 13: Function to create Interrupt

extern bit EE_RSCAN_SetInterrupt(u08 Unit_u08, u08 Channel_u08, u08 InterruptSelection_u08, u16 InterruptSubSelection_u16);				
Description:	*This function enables the specific Interrupt or Sub Interrupt Element of the Interrupt Line.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	InterruptSelection_u08	*Unsigned Char*	*Interrupt Source*	*EE_RSCAN_INT_GERR*
				EE_RSCAN_INT_RXF0
				EE_RSCAN_INT_RXF1
				EE_RSCAN_INT_TX
				EE_RSCAN_INT_TXA
				EE_RSCAN_INT_TXQ
				EE_RSCAN_INT_CERR
				EE_RSCAN_INT_TXHL
				EE_RSCAN_INT_RXCF
				EE_RSCAN_INT_TXCF
	InterruptSubSelection_u16	*Unsigned Char*	*Enable Interrupt Element*	*Depends on InterruptSelection_u08*
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 14: Function to set Interrupt

extern bit EE_RSCAN_GetStatus(u08 Unit_u08, u08 Channel_u08, u08 StatusNumber_u08, pu08 StatusValue_u08);				
Description:	*This function checks the Global or Channel status*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
				EE_RSCAN_GLOBAL
	StatusNumber_u08	*Unsigned Char*	*Status to be checked*	*EE_RSCAN_STATUS_OPMODE*
				EE_RSCAN_STATUS_PSMODE
				EE_RSCAN_STATUS_RECEIVE
				EE_RSCAN_STATUS_TRANSMIT
				EE_RSCAN_STATUS_BUSOFF
				EE_RSCAN_STATUS_ERRCNTLEV
				EE_RSCAN_STATUS_TOVF
				EE_RSCAN_STATUS_THPM
				EE_RSCAN_STATUS_TGPT
				EE_RSCAN_STATUS_NEWTXHISTORY
				EE_RSCAN_STATUS_VALID
				EE_RSCAN_STATUS_TRERRCOUNT
				EE_RSCAN_STATUS_RXERRCOUNT
				EE_RSCAN_STATUS_ERRPLEV
				EE_RSCAN_STATUS_INT_RXFIFO
				EE_RSCAN_STATUS_INTERRUPT
	StatusValue_u08	*Unsigned Char*	*Status Flag/Value*	*Depends on StatusNumber*
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 15: Function to get Status

extern bit EE_RSCAN_GetError(u08 Unit_u08, u08 Channel_u08, pu16 InterruptErrorFlag_pu16, pu16 LastErrorFlag_pu16);				
Description:	*This function gives the Global or Channel depended Error Source back. The Information contains the Last Occurred Error Code and Last Error Flag.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
				EE_RSCAN_GLOBAL
	InterruptErrorFlag_pu16	*Unsigned Short Pointer*	*Last occurred Error Code*	*EE_RSCAN_GINT_NONE*
				EE_RSCAN_GINT_MSGLOST
				EE_RSCAN_GERROR_THLLOST
	LastErrorFlag_pu16	*Unsigned Char*	*Status Flag/Value*	*GERFLERR*
				ERFLERR
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 16: Function to get Errors

extern bit EE_RSCAN_SetGlobalConfiguration(u08 Unit_u08, const struct ee_rscan_cfg_global Config);*				
Description:	*This Function configures the Global Macro Settings.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Config	*Const struct ee_rscan_cfg_global**	*Global Configuration of the RSCAN Macro.*	*Global Configuration*
				Global Control
				RMNB
				RNC
				RFCC
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 17: Function to set Global Configuration

extern bit EE_RSCAN_SetChannelConfiguration(u08 Unit_u08, u08 Channel_u08, const struct ee_rscan_cfg_global Config);*				
Description:	*This Function configures the Channel Macro Settings.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Config	*Const struct ee_rscan_cfg_global**	*Global Configuration of the RSCAN Macro.*	*Global Configuration*
				Global Control
				RMNB
				RNC
				RFCC
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 18: Function to set Channel Configuration

*extern bit EE_RSCAN_SendMessage(u08 Unit_u08, u08 Channel_u08, u08 *Status_u08, struct ee_rscan_message* Message);*				
Description:	*This Function sends the Message for a specific Channel and Transmit Buffer.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Message	*Struct ee_rscan_message**	*Message to be send*	*Header*
				Flag (ts, ptr and dlc)
				Data
				Path
				Pathdetail
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 19: Function to send Message

extern bit EE_RSCAN_TxAbort(u08 Unit_u08, u08 Channel_u08, struct ee_rscan_message Message);*				
Description:	*This Function aborts the transmission Process of Message Box and RX/TX FIFO.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Message	*Struct ee_rscan_message**	*Message to be send*	*Header*
				Flag (ts, ptr and dlc)
				Data
				Path
				Pathdetail
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 20: Function to abort Transmission

extern bit EE_RSCAN_ReceiveMessage(u08 Unit_u08, u08 Channel_u08, pu08 Status_pu08, struct ee_rscan_message Message);*				
Description:	*This Function is used to receipt the Message addressed at Message_Path. For detailed selection of Message_Path, please consider the Message-Data Description.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	Message	*Struct ee_rscan_message**	*Message to be send*	*Header*
				Flag (ts, ptr and dlc)
				Data
				Path
				Pathdetail
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 21: Function to receive Message

extern bit EE_RSCAN_SetAFLEntry(u08 Unit_u08, u08 Channel_u08, u16 RuleNumber_u16, struct ee_rscan_a_afl AFLEntry);				
Description:	*This Function enters a new entry into the AFL.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Channel_u08	*Unsigned Char*	*Specific Channel at Macro*	*0*
	RuleNumber_u16	*Unsigned Short*	*Entry Number to be entered*	*0 … 16*
	AFLEntry	*Struct ee_rscan_a_afl*	*AFL Entry (Rule)*	*id*
				mask
				ptr0
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 22: Function to set one item in the Acceptance Filter Rule Table

extern bit EE_RSCAN_Enable_RXFIFO(u08 Unit_u08, u08 RXFIFO_EN_u08);				
Description:	*Temporary Function for Application Test. Enables RX FIFO in Operation Mode.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
	Unit_u08	*Unsigned Char*	*RX FIFO to be enabled*	*EE_RSCAN_GL_CONFIG_RXFIF0_EN0*
				EE_RSCAN_GL_CONFIG_RXFIF0_EN1
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 23: Function to enable Receive FiFo

extern bit EE_RSCAN_RAMTest(u08 Unit_u08);				
Description:	*Temporary Function for Application Test. This Function Test the hole available RAM for this Macro.*			
Direction:	Parameter:	Type:	Description:	Value:
Input	*Unit_u08*	*Unsigned Char*	*Specific RSCAN Macro at Platform*	*0*
Output	*Return*	*Bit*	*Return value gives a true or false back. It's a function error check.*	*True*
				False

Table 24: Function to execute RAM Test

Chapter 4 Example for using RSCAN-Lite SW Driver

The Examples could be found at the files “rscan_a.c”, “rscan_a.h” and “rscan_s.h”.

The List of the Tests and Applications are shown below:

Application 0 Func.:	*EE_RSCAN_COM_FIFO_TX_and_RX_FIFO_Test*

Description:	● Test the TX/RX FIFO (COM FIFO) for one Channel
	● Test RX FIFO for one Channel
	● Transmit Test Messages with the TX/RX FIFO
	  ○ Push Messages into the FIFO
	  ○ Check TX/RX FIFO Full Flag
	● Receipt Test Messages with RX FIFO
	● Create and Test Interrupts for Transmission and Reception

Application 1 Func.:	*EE_RSCAN_TX_RX_MB_Test*

Description:	● Test the TX Message Buffer for one Channel
	● Test the RX Message Buffer for one Channel
	● Create Interrupt for Transmission
	● Transmit Test Message over TX Message Buffer
	  ○ Push Message into Buffer
	  ○ Wait till Transmission Interrupt Occur
	● Receipt Messages in RX Message Buffer

Application 2 Func.:	*EE_RSCAN_TX_RX_MB_Loop_Test*
Description:	● Same Test as “Application 1” ● Communication over intern Loop Back

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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord 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, 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 one with a different part number, confirm that the change will not lead to problems.

   > The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.

6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

“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.
“High Quality”:	Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti- crime systems; safety equipment; and medical equipment not specifically designed for life support.
“Specific”:	Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

Sales Offices

Renesas Electronics Corporation

www.renesas.com

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

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

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

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, 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.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

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

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

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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

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