EPSON
EXCEED YOUR VISION

S1C17801
REMC Application Note

When using the commands, follow the instructions of NOTICE_Application Notes Sample Programs.pdf being included in the downloaded compressed file.

SEIKO EPSON CORPORATION

Rev.1.0

NOTICE

OVERVIEW

This document is a reference to use the REMC function of S1C17801.

OPERATING ENVIRONMENT

- S5U1C17801T1100 (hereafter SVT17801:Software eValuation Tool for S1C17801）
  A set of SVT17801 CPU board and SVT17 ICD board (two sets as necessary)
- USB miniB cable
- PC
  The GNU17 development tool has been installed.
  Installed with the USB driver for the SVT17 ICD board

Table of Contents

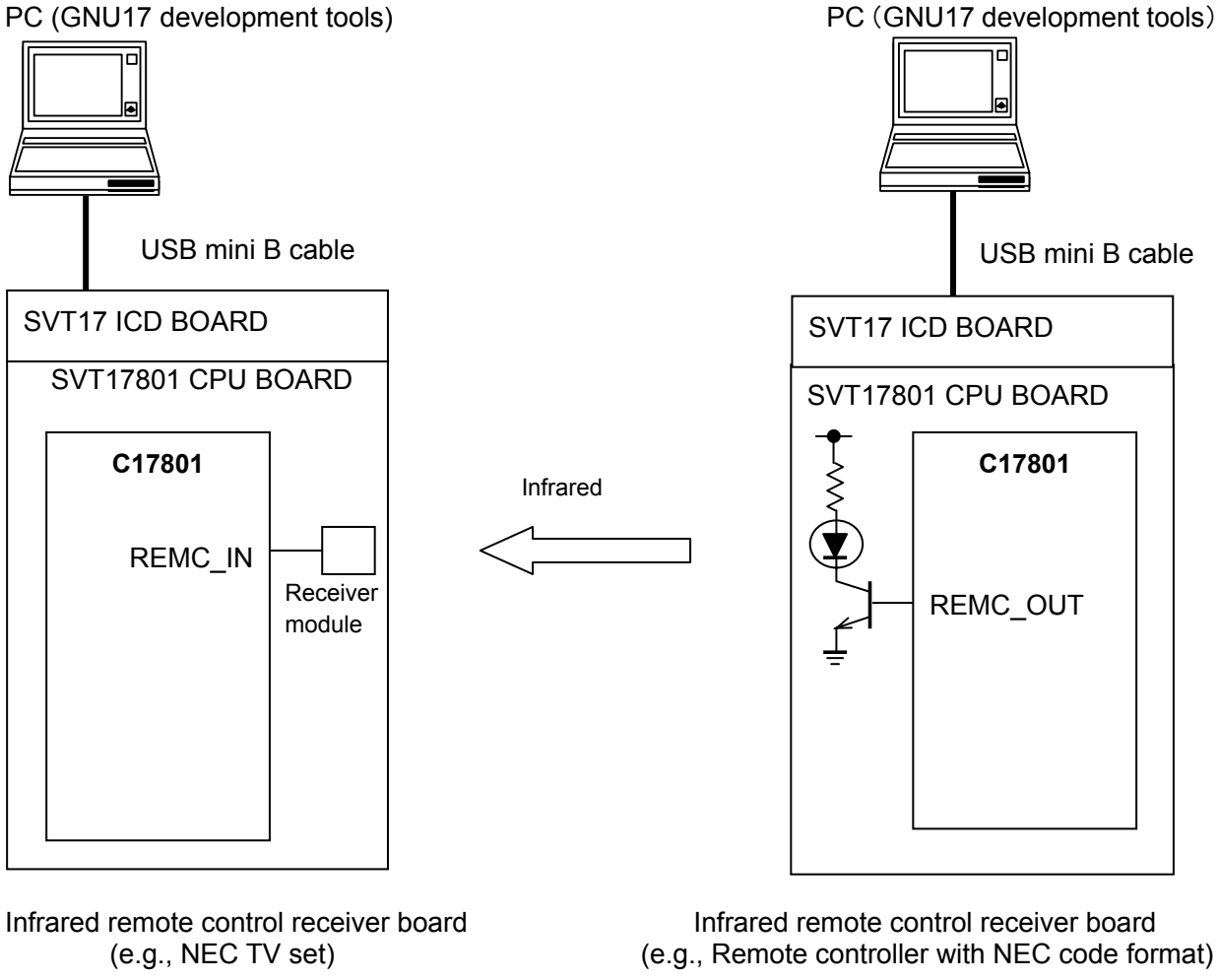
1. SPECIFICATIONS

This sample software executes the NEC formatted transmit/receive process. (Refer to *Note)
Panasonic PNA4702M is used for the infrared receiver module.
Stanley AN333 is used for the infrared light emitting diode.
For more information, refer to each data sheet of the relevant product.

A prompt “command > ”appears in the [Simulated I/O] window. The following table shows available commands.

Command list

Command		Description
tran		Executes transmission process
	pow	Transmits the power button code
	mute	Transmits the mute button code
	1 to 12	Transmits 1 to 12 key codes
	up	Transmits the volume turn-up button code
	down	Transmits the volume turn-down button code
	exit	Exits the transmission process
recv		Receives data transmitted from a remote controller and displays the code. Exits the receive process at the receipt of the power button code.



Infrared remote control receiver board
(e.g., NEC TV set)

Infrared remote control receiver board
(e.g., Remote controller with NEC code format)

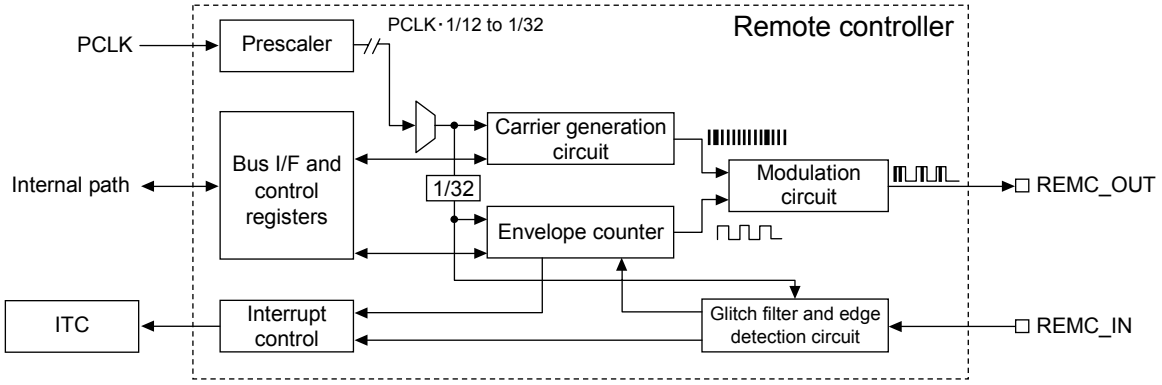
2. DESCRIPTIONS OF FUNCTIONS USED

Input and output pins

Receiver module (U21) and infrared LED (D1) are connected with REMC_IN and REMC_OUT respectively. This port is shared with P52 and P53.

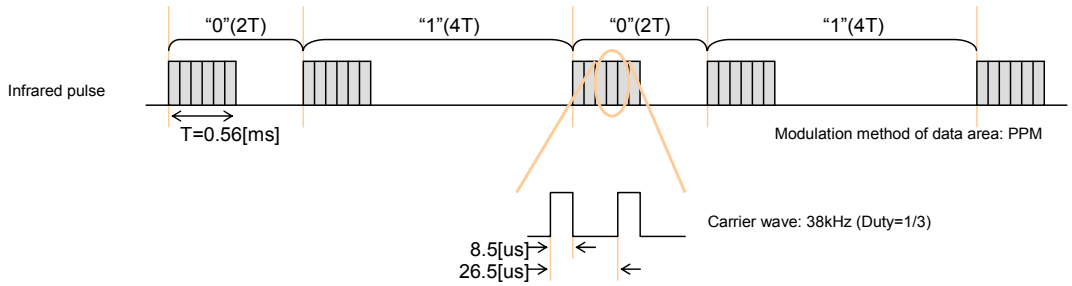
Interrupt

The REMC interrupt vector number and the vector address are as follows:
Vector number:30 (0x1e)
Vector address: 0x900078
The REMC module can generate two types of interrupts, i.e., transmit interrupt and receive interrupt.



3. OPERATION

The following shows the overview of infrared pulse.



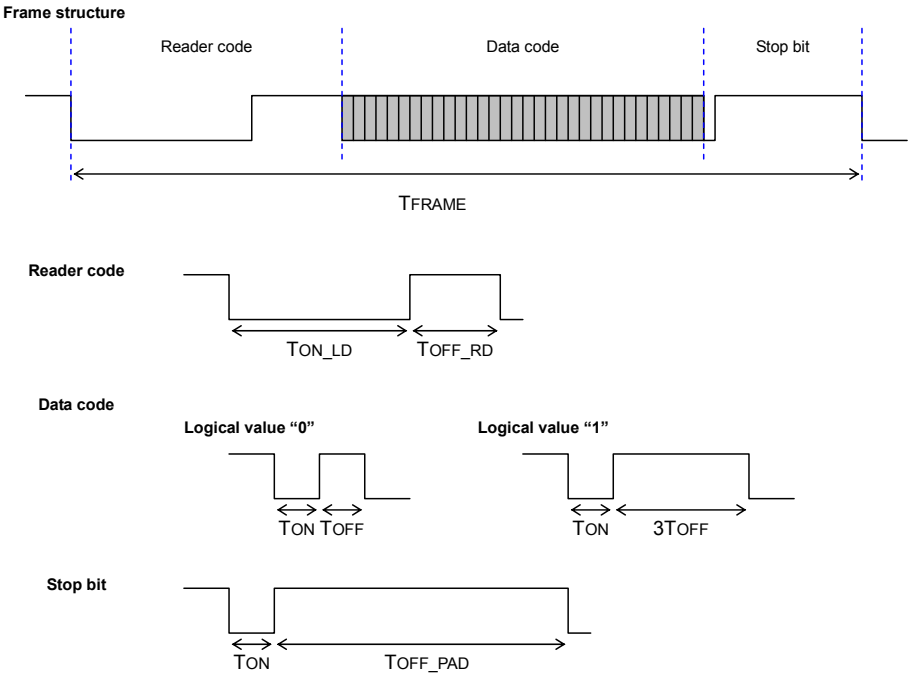
3.1 NEC format

Name	Description	Value	Remark
Time unit T	T =TON =TOFF	0.56[ms]	
Reader code	TON_LD	16T	
	TOFF_LD	8T	
Stop bit	TON	1T	
	TOFF_PAD	—	30 [ms] or longer
Frame length	TFRAME	108[ms]	
Data code	Custom code	16[bit]	
	Data code	8[bit]	
	Reverse bit of data code	8[bit]	

3.2 AEHA (Association for Electric Home Appliances) format

Name	Description	Value	Remark
Time unit T	T =TON =TOFF	0.35 - 0.50 [ms]	Typically 0.47 [ms]
Reader code	TON_LD	8T	
	TOFF_LD	4T	
Stop bit	TON	1T	
	TOFF_PAD	8 [ms] or longer	
Frame length	TFRAME	—	Up to approx. 160 [ms]
Data code	Custom code	16 [bit]	Up to 9 [Byte]
	Parity	4 [bit]	
	Data code	4 [bit]	
	Data	48 [bit] or less	

3.3 Frame structure



An infrared signal has a frame structure consisting of reader code, data code and stop bit areas.

[Reader code]

An ON pulse of T_{ON_LD} or longer is followed by an OFF pulse of T_{OFF_LD}. The pulse length is regulated by each AEHA format.

[Data code]

One bit is indicated by a pair of ON → OFF pulses. If an ON pulse of T_{ON} is followed by an OFF pulse of T_{OFF}, then the logical value is "0". If an ON pulse of T_{ON} is followed by an OFF pulse of $3T_{OFF}$, then the logical value is "1".

Bits are ordered from LSB to MSB (e.g., 0x12 becomes 01001000).

The total number of bits in a data code is regulated by each AEHA format.

[Stop bit]

An ON pulse of T_{ON} followed by an OFF pulse of T_{OFF_PAD} after the end of a data code indicates a stop bit.

An OFF pulse area tends to be used as a frame padding. It means that the frame length T_{FRAME} is kept constant typically by adjusting the length of this T_{OFF_PAD}.

4. SOFTWARE DESCRIPTION

4.1 File Configuration

File name	Function
boot.c	Startup module
inthdlr.c	Interrupt handler
main.c	REMC sample program main process
vector.c	Vector table
header/reg_801.h	Register definitions
header/vector.h	Vector table definitions
remc_drv/remc_drv.c	REMC driver API group
remc_drv/remc_api.h	REMC driver API definitions
cmu_drv/	CMU driver folder
flashc_drv/	FLASHC driver folder
gpio_drv/	GPIO driver folder
itc_drv/	ITC driver folder
sramc_drv/	SRAMC driver folder
remc_gnu17IDE.lds	Linker script file
remc_gnu17IDE.cmd	GDB command file
remc_gnu17IDE.par	Parameter configuration file
remc_gnu17IDE.mak	Makefile
.cdtproject	Project file
.gnu17project	Project file
.project	Project file
GDB17 Launch for remc.launch	Project startup file

4.2 Descriptions of the Modules

File name：main.c

Function name	Function
main	Configures Port/ITC/REMC and interrupts, and transmits/receives REMC.
RemcInitialize	Initializes the transmission/receipt of REMC.
PortInitialize	Sets ports for REMC.
ItcInitialize	Sets REMC interrupts.
InterruptMaskREMC	Enable/Disable REMC interrupts.

For ITC driver functions, refer to the “Detailed description of ITC driver” chapter.
For the GPIO driver functions, refer to the “Detailed description of GPIO driver” chapter.
For the CMU driver functions, refer to the “Detailed description of CMU driver” chapter.
For the FLASHC driver functions, refer to the “Detailed description of FLASHC Driver” chapter.
For SRAMC driver functions, refer to “Detailed descriptions of SRAMC driver” chapter.

4.3 Global Variables

The following shows the global variables used in the sample program.

Variable name	Type	Function
remBuffer[5]	unsigned char	Buffer for setting and analyzing transmit/receive commands.
inbuff[128]	unsigned char	Received data buffer
outbuff[128]	unsigned char	Transmit data buffer

4.4 Structure

Definition name		
T_REMC_CFG	config	
Member		
mode	unsigned short	Transmit/Receive mode
systemClock	unsigned long	System clock
divide	unsigned short	Prescaler divide ratio
carrierHigh	unsigned short	Carrier High
carrierLow	unsigned short	Carrier Low
dataCodeBit	unsigned short	Bit count of Data code
T_ON	unsigned short	T ON pulse wide (us)
T_OFF	unsigned short	T OFF pulse wide(us)
averageError	unsigned short	average error(us)
leaderON	unsigned short	Leader Code ON (T_ON count)
leaderOFF	unsigned short	Leader Code OFF (T_OFF count)
data0ON	unsigned short	Bit data 0 ON (T_ON count)
data0OFF	unsigned short	Bit data 0 OFF (T_OFF count)
data1ON	unsigned short	Bit data 1 ON (T_ON count)
data1OFF	unsigned short	Bit data 1 OFF (T_OFF count)
stopON	unsigned short	Stop Code ON (T_ON count)
stopOFF	unsigned short	Stop Code OFF (T_OFF count)
intMaskFunc	void(*)(unsigned char)	Interrupt mask function
Remarks		
Structure for setting the initial value of REMC.		

4.5 Operation Procedure

Import the project
(1) Launch the IDE and import the "remc" project.
* For information on importing projects, refer to "3. Software Development Procedures" in S5U1C17001C Manual.

Build
(1) Build the "remc" project by using the IDE.

Connection and powering on procedures
(1) Connect SVT17801, USB miniB cable and PC.
(2) Reset SVT17 ICD board.

Run the program
(1) Execute "remc" project by using the IDE.
(2) After executing "remc", a prompt "command >" is displayed in the [Simulated I/O] window.
(3) Input one of the following commands.

Command list

Command		Description
tran		Executes transmission process
	pow	Transmits the power button code
	mute	Transmits the mute button code
	1 to 12	Transmits 1 to 12 key codes
	up	Transmits the volume turn-up button code
	down	Transmits the volume turn-down button code
	exit	Exits the transmission process
recv		Receives data transmitted from a remote controller and displays the code. Exits the receive process at the receipt of the power button code.

Inputting the "tran" command activates the remote transmission process and displays a prompt "tran command > " in the [Simulated I/O] window. Inputting each "pow", "mute", "1 to 12", "up" or"down" command from the "trans command > " prompt transmits a corresponding key code from the controller. Input "exit" to exit the transmission process.

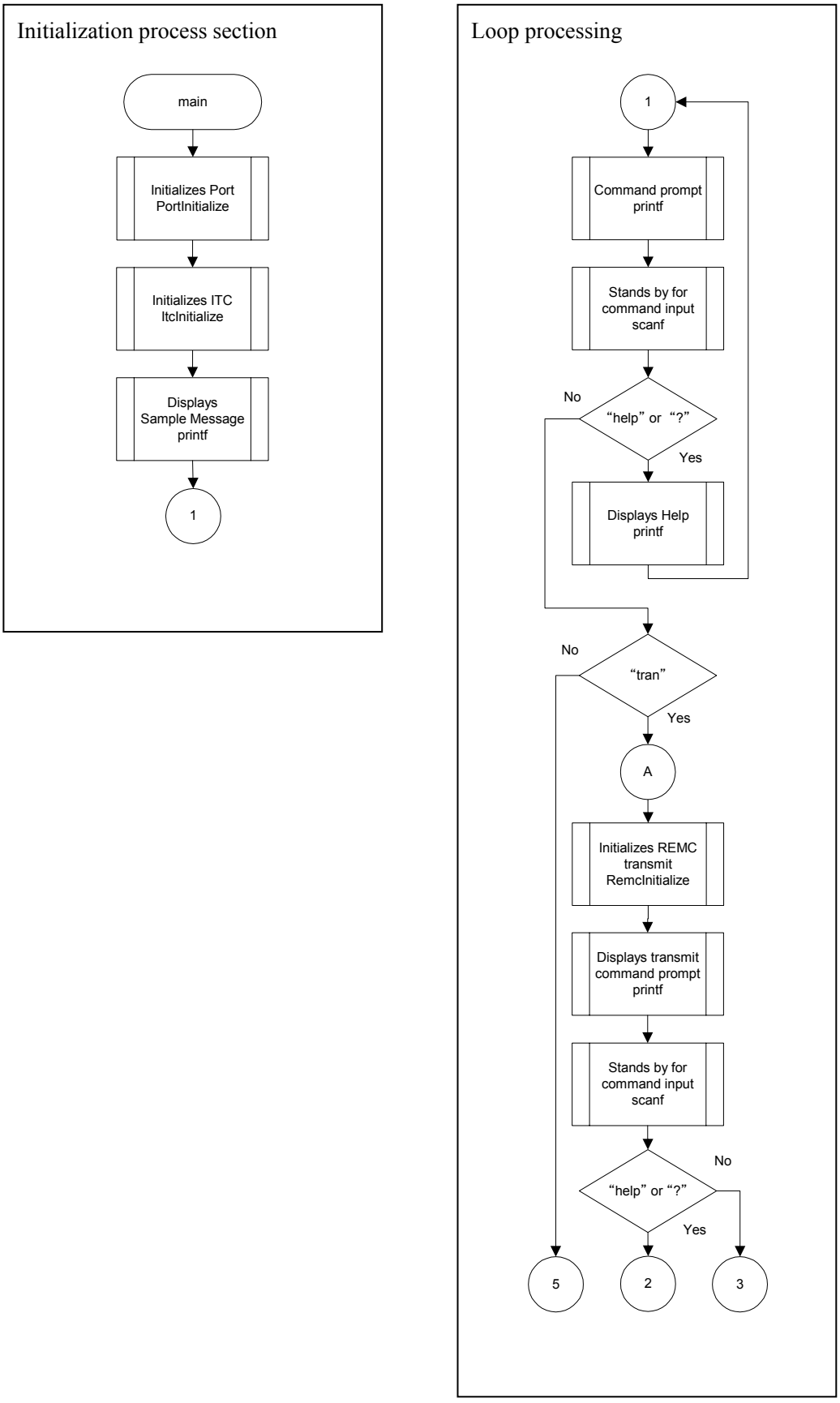
Inputting the "recv" command activates a remote receive process. The controller receives the input key code from an external remote controller, and outputs the key code into the [Simulated I/O] window. The remote receive process terminates when the controller receives the power button code from the external remote controller.

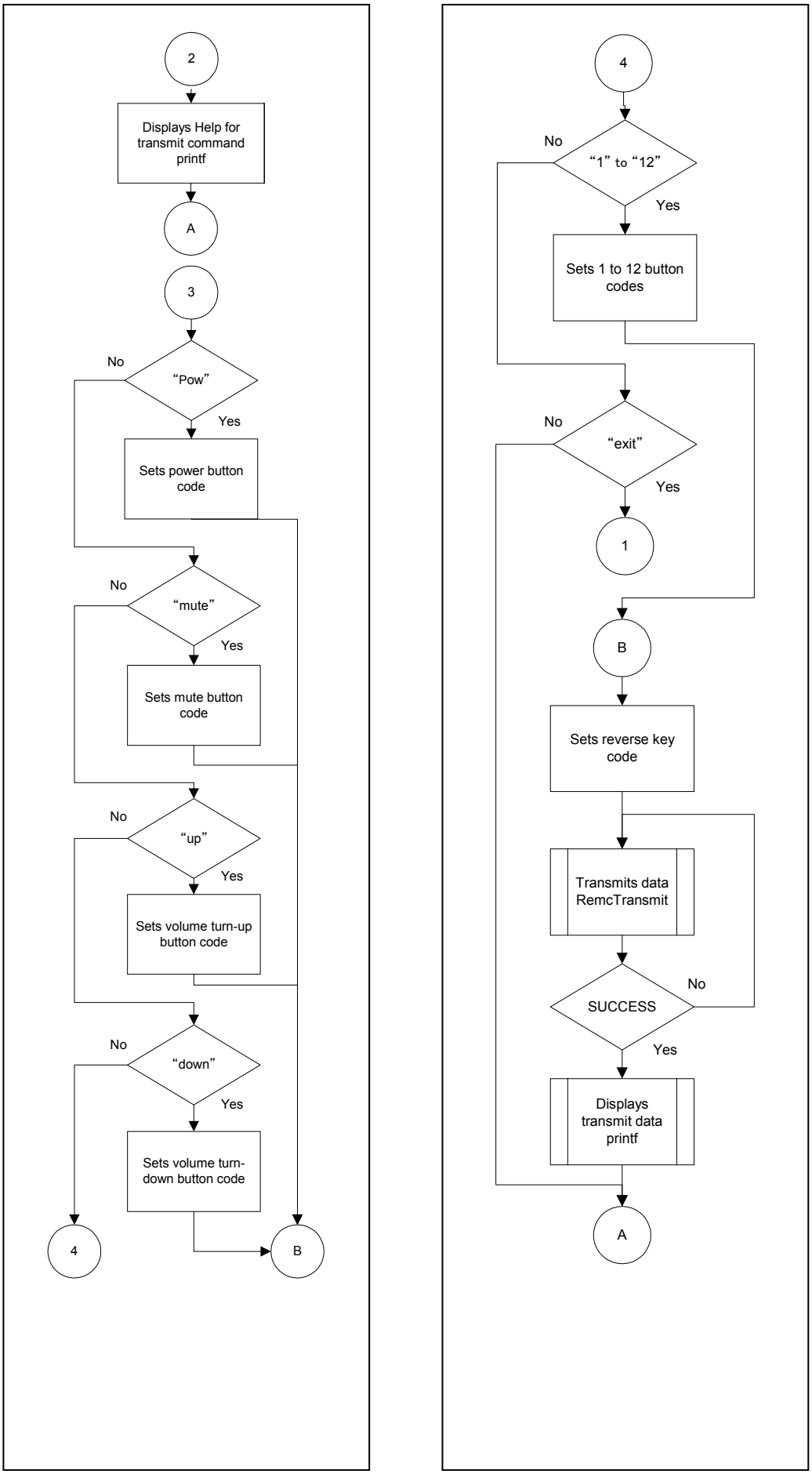
4.6 Outline of the sample program operations

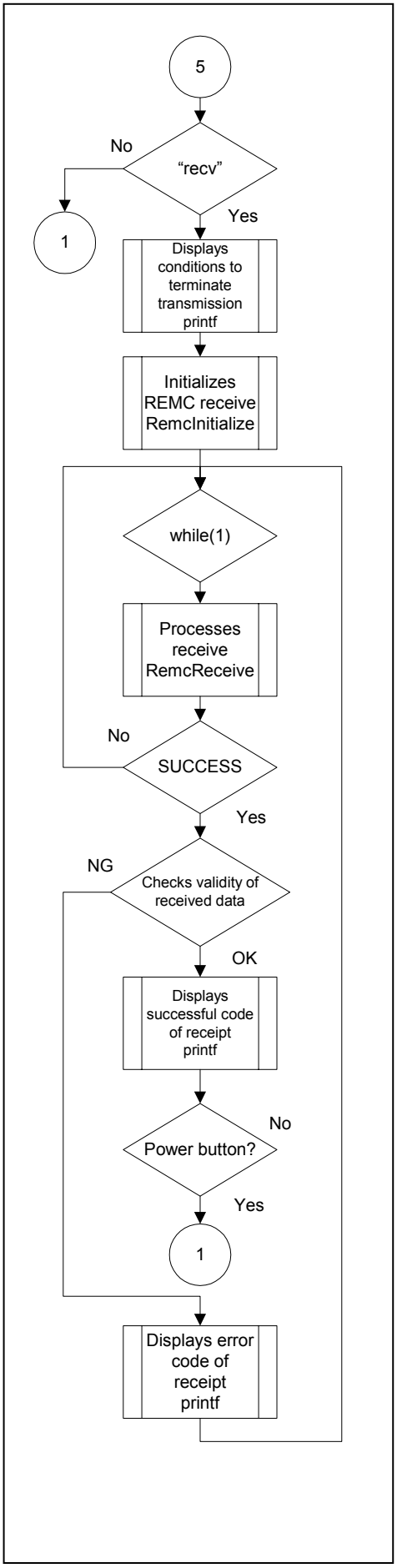
(1) Initializes the port.
(2) Initializes the ITC.
(3) Enables the CPU interrupt handler.
(4) Stands by for command input from the Simulated I/O
(5) Initializes REMC either for transmission or for receipt depending on an input command.
(6) Starts the relevant process as set in the step (5).

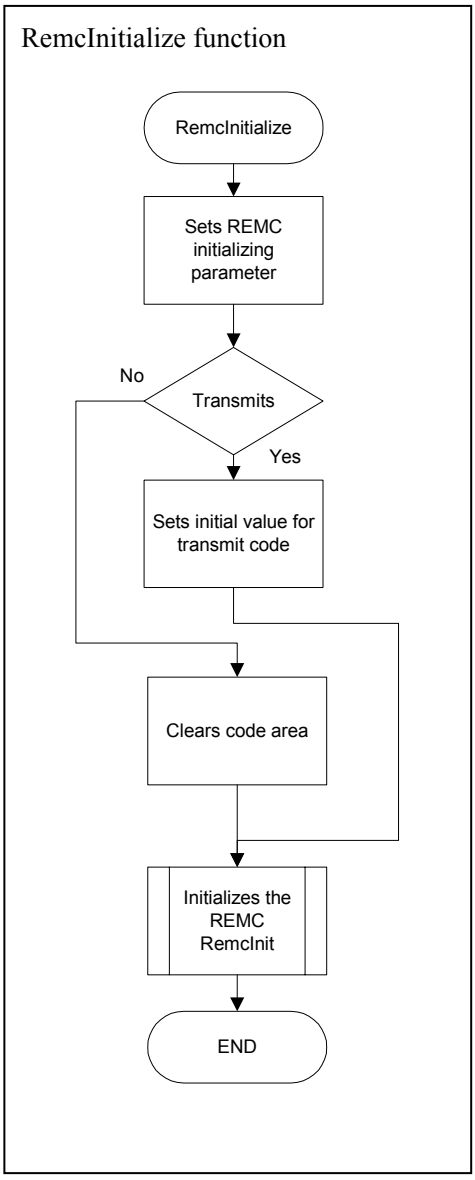
4.7 Flowchart

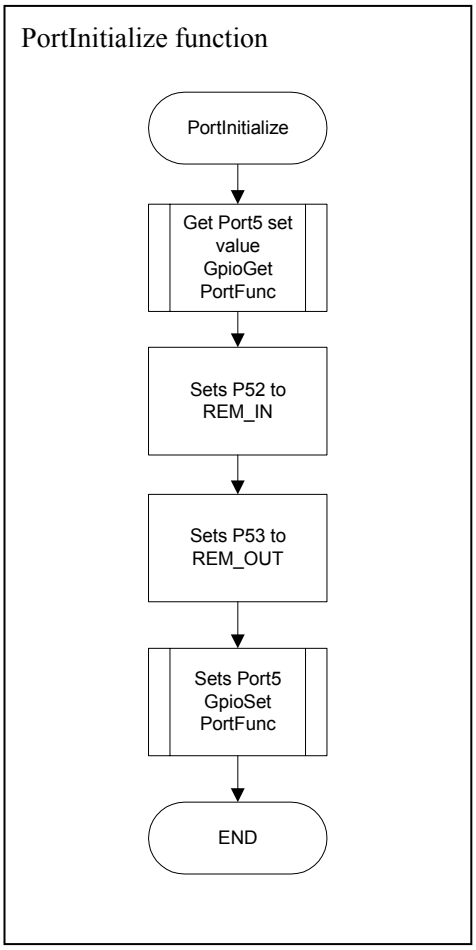
The following shows the flowchart of the main routine and interrupt handler functions.

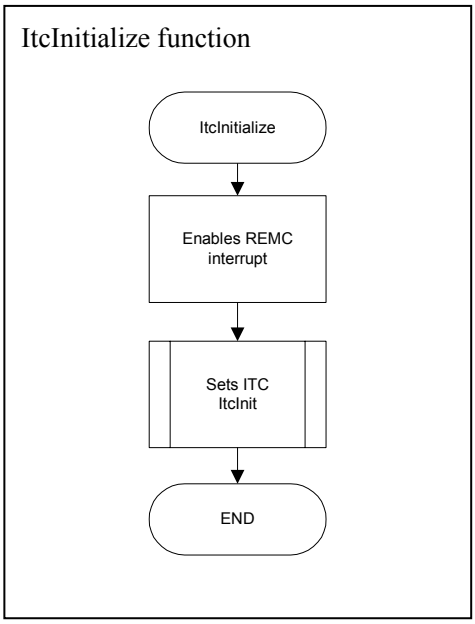


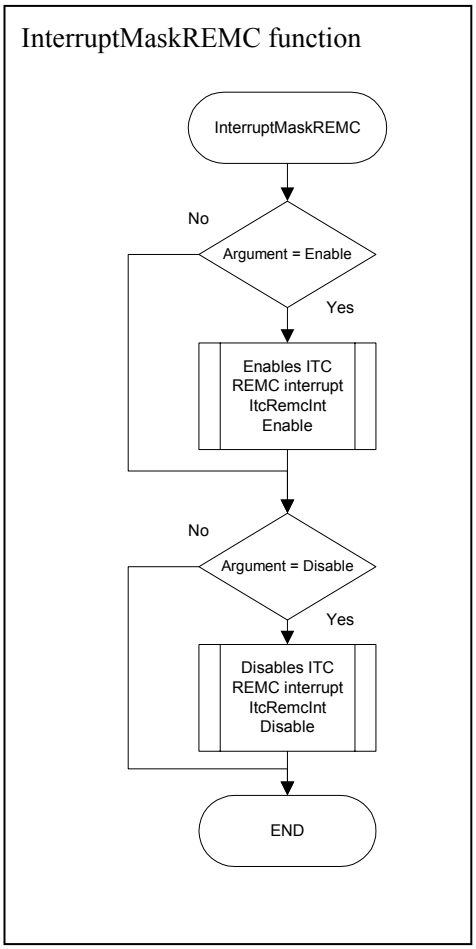

2
Displays Help for transmit command printf
A
3
No
"Pow"
Yes
Sets power button code
No
"mute"
Yes
Sets mute button code
No
"up"
Yes
Sets volume turn-up button code
No
"down"
Yes
Sets volume turn-down button code
4
B
4
No
"1" to "12"
Yes
Sets 1 to 12 button codes
No
"exit"
Yes
1
B
Sets reverse key code
Transmits data RemcTransmit
No
SUCCESS
Yes
Displays transmit data printf
A


5
No
"recv"
Yes
1
Displays conditions to terminate transmission printf
Initializes REMC receive RemcInitialize
while(1)
Processes receive RemcReceive
No
SUCCESS
Yes
NG
Checks validity of received data
OK
Displays successful code of receipt printf
No
Power button?
Yes
1
Displays error code of receipt printf


RemcInitialize function
RemcInitialize
Sets REMC initializing parameter
No
Transmits
Yes
Sets initial value for transmit code
Clears code area
Initializes the REMC RemcInit
END


PortInitialize function
PortInitialize
Get Port5 set value GpioGet PortFunc
Sets P52 to REM_IN
Sets P53 to REM_OUT
Sets Port5 GpioSet PortFunc
END


ItcInitialize function
ItcInitialize
Enables REMC interrupt
Sets ITC ItcInit
END


InterruptMaskREMC function
InterruptMaskREMC
No
Argument = Enable
Yes
Enables ITC REMC interrupt ItcRemcInt Enable
No
Argument = Disable
Yes
Disables ITC REMC interrupt ItcRemcInt Disable
END

4.8 Detailed Explanation of the REMC driver

The following provides descriptions on the functions described in the files memc_drv.c and memc_api.h.

4.8.1 REMC driver low level API

Setting REMC Prescaler

Format	int RemcSetPrescale(unsigned char divider)
Function	Sets REMC Prescaler
Argument	divider – in Division ratio REMC_PSDIV_32 : PCLK/32 REMC_PSDIV_24 : PCLK/24 REMC_PSDIV_16 : PCLK/16 REMC_PSDIV_12 : PCLK/12
Return value	SUCCESS: Terminated successfully ERROR : Undefined
(Description) Sets Prescaler with the specified division ratio. If Prescaler is in ON status, ERROR is returned.	

Obtaining REMC Prescaler (Macro function)

Format	unsigned char RemcGetPrescale(void)
Function	Obtains REMC Prescaler
Argument	None
Return Value	REMC_PSDIV_32: PCLK/32 REMC_PSDIV_24: PCLK/24 REMC_PSDIV_16: PCLK/16 REMC_PSDIV_12: PCLK/12
(Description) Returns the operating mode currently set.	

Starting REMC Prescaler (Macro function)

Format	void RemcStartPrescale(void)
Function	Starts REMC Prescaler
Argument	None
Return Value	None
(Description) Starts Prescaler	

Stopping REMC Prescaler (Macro function)

Format	void RemcStopPrescale(void)
Function	Stops REMC Prescaler
Argument	None
Return Value	None
(Description) Stops Prescaler	

Obtaining operating status of REMC Prescaler (Macro function)

Format	unsigned char RemcStatePrescale(void)
Function	Obtains operating status of REMC Prescaler
Argument	None
Return Value	REMC_PS_ON: Operating REMC_PS_OFF: Stop
(Description) Returns the operating status of Prescaler.	

Setting REMC operating mode (Macro function)

Format	void RemcSetMode(unsigned short mode)
Function	Sets REMC operating mode
Argument	mode – in Operating mode REMC_TRANSMIT : Transmit mode REMC_RECEIVE : Receive mode
Return Value	None
(Description) Sets REMC operating mode.	

Obtaining REMC operating mode (Macro function)

Format	unsigned short RemcGetMode(void)
Function	Obtains REMC operating mode
Argument	None
Return Value	REMC_TRANSMIT : Transmit mode REMC_RECEIVE : Receive mode
(Description) Returns REMC operating mode.	

Enabling REMC underflow interrupt (Macro function)

Format	void RemcEnableUnderflow(void)
Function	Enables the REMC underflow interrupt
Argument	None
Return Value	None
(Description) Enables the underflow interrupt handler of Envelope Counter under transmit mode.	

Disabling REMC underflow interrupts (Macro function)

Format	void RemcDisableUnderflow(void)
Function	Disables the REMC underflow interrupt
Argument	None
Return Value	None
(Description) Disables the underflow interrupt handler of Envelope Counter under transmit mode.	

Enabling REMC falling edge interrupts (Macro function)

Format	void RemcEnableFalling(void)
Function	Enables the REMC falling edge interrupt
Argument	None
Return Value	None
(Description) Enables the falling edge interrupt handler under receive mode.	

Disabling REMC falling edge interrupts (Macro function)

Format	void RemcDisableFalling(void)
Function	Disables the REMC falling edge interrupt
Argument	None
Return Value	None
(Description) Disables the falling edge interrupt handler under receive mode.	

Enabling REMC rising edge interrupt (Macro function)

Format	void RemcEnableRising(void)
Function	Enables the REMC rising edge interrupt
Argument	None
Return Value	None
(Description) Enables the rising edge interrupt handler under receive mode.	

Disabling REMC rising edge interrupt (Macro function)

Format	void RemcDisableRising(void)
Function	Disables the REMC rising edge interrupt
Argument	None
Return Value	None
(Description) Disables the rising edge interrupt handler under receive mode.	

Starting REMC operation (Macro function)

Format	void RemcStartREMC(void)
Function	Starts REMC operation
Argument	None
Return Value	None
(Description) Starts REMC operation with the configured operating mode.	

Stopping REMC operation (Macro function)

Format	void RemcStopREMC(void)
Function	Stops REMC operation
Argument	None
Return Value	None
(Description) Stops REMC operation with the configured operating mode.	

Setting REMC carrier wave

Format	void RemcSetCarrier(unsigned char high, unsigned char low)
Function	Sets REMC carrier wave
Argument	High – in Sets the length of high output 0 to 255 Low – in Sets the length of low output 0 to 255
Return Value	None

(Description)
Sets the specified carrier wave.

Prescaler clock frequency	2MHz		
Carrier frequency	30kHz	38kHz	56kHz
Carrier duty	1/2	1/3	1/4
Carrier cycle time	33.333μs	26.316μs	17.857μs
Carrier high load register	32	17	8
Carrier low load register	32	34	26

Setting REMC transmit waveform width (Macro function)

Format	void RemcSetTransmit(unsigned short envelope)
Function	Sets the width of REMC transmit waveform
Argument	Envelope – in Sets the width of transmit waveform 0 to 65535
Return Value	None
(Description) Sets the specified waveform width for transmission.	

Obtaining REMC receive waveform width (Macro function)

Format	unsigned short RemcGetReceive(void)
Function	Sets the width of REMC transmit waveform
Argument	None
Return Value	Receive waveform width: 0 to 65535
(Description) Returns the received waveform width.	

4.8.2 **Details on REMC high level API**

Initializing the REMC

Format	void RemcInit(T_REMC_CFG *config)
Function	Initializes the REM module.
Argument	config - in Initialization data
Return Value	None

(Description)
Initializes the REM module and REMC process.

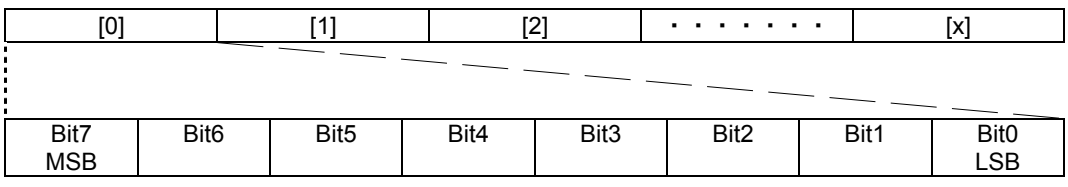
```
typedef struct {
        unsigned short      mode;           // Transmit and receive mode
                                            // REMC_TRANSMIT   : Transmit mode
                                            // REMC_RECEIVE    : Receive mode
        unsigned short      systemClock;    // System clock value
        unsigned short      divide;         // Prescaler division ratio
                                            // REMC_PSDIV_32     : PCLK/32
                                            // REMC_PSDIV_24     : PCLK/24
                                            // REMC_PSDIV_16     : PCLK/16
                                            // REMC_PSDIV_12     : PCLK/12
        unsigned short      carrierHigh;    // Carrier High
        unsigned short      carrierLow;     // Carrier Low

        // Infrared-Red Remote Controller format
        unsigned short      dataCodeBit;    // Data code bit numbers

        unsigned short      T_ON;           // T ON pulse wide (us)
        unsigned short      T_OFF;          // T OFF pulse wide (us)

        unsigned short      averageError;   // Average error(us)

        unsigned short      leaderON;       // Leader Code ON   (T_ON count)
        unsigned short      leaderOFF;      // Leader Code OFF (T_OFF count)
        unsigned short      data0ON;        // Bit data 0 ON    (T_ON count)
        unsigned short      data0OFF;       // Bit data 0 OFF (T_OFF count)
        unsigned short      data1ON;        // Bit data 1 ON    (T_ON count)
        unsigned short      data1OFF;       // Bit data 1 OFF (T_OFF count)
        unsigned short      stopON;         // Stop Code ON    (T_ON count)
        unsigned short      stopOFF;        // Stop Code OFF (T_OFF count)
}   T_REMC_CFG;
```

Transmitting REMC data

Format	int RemcTransmit(unsigned char *buffer)
Function	Data transmission
Argument	buffer – in Storage pointer to transmit data
Return Value	SUCCESS : Terminated successfully PROCESSING: Processing

(Description)
Transmits data of the specified buffer.
Transmit data size is the number of bits specified by Config.dataCodeBit.

buffer

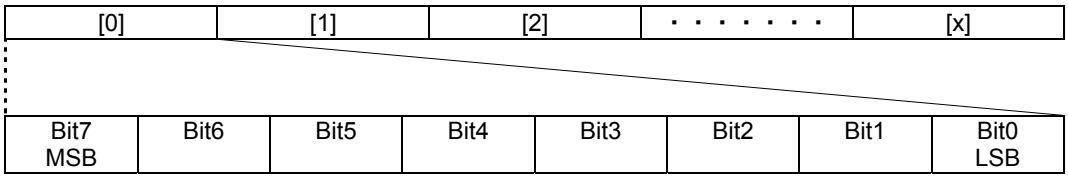


Receiving REMC data

Format	int RemcReceive(unsigned char *buffer)
Function	Receives data
Argument	buffer – out Storage pointer to receive data
Return Value	SUCCESS : Terminated successfully PROCESSING: Processing

(Description)
Stores the received data to buffer.
If SUCCESS is returned, the buffer data becomes valid. If PROCESSING is returned, call this function again.
Receive data size is the number of bits specified by Config.dataCodeBit.

buffer



Stopping the REMC

Format	void RemcStop(void)
Function	Stops the REMC
Argument	None
Return Value	None

(Description)
Stops the REMC. The function stops Prescaler and REMC operations, and clears internal variables.

Capturing REMC interrupt handler

Format	void RemcInterruptHandler(void)
Function	Capturing interrupt handler
Argument	None
Return Value	None

(Description)
Capturing interrupt flags when sending and receiving data.

Processing REMC interrupt

Format	void RemcInterruptProcess(void)
Function	Processes interrupt handlers
Argument	None
Return Value	None

(Description)
Processes transmission and receipt of data based on the reaped interrupt flags.

4.9 Header Definitions

The table below show the definitions used in the driver functions.

Following is the definitions for internal operation.

Definition name	Value	Description
STATE_IDLE	0	Wave state in IDLE
STATE_DUMMY	1	Wave state in DUMMY
STATE_LEADER_H	2	Wave state in leader high
STATE_LEADER_L	3	Wave state in leader low
STATE_DATA_H	4	Wave state in data high
STATE_DATA_L	5	Wave state in data low
STATE_STOP_H	6	Wave state in stop high
STATE_STOP_L	7	Wave state in stop low
STATE_COMP	8	Wave state complete

Following is the definitions used in API.

Definition name	Value	Description
REMC_TRANSMIT	0x0000	Transmit mode
REMC_RECEIVE	0x0080	Receive mode
REMC_PSDIV_32	0x0003	Clock Divide PCLK/32
REMC_PSDIV_24	0x0002	Clock Divide PCLK/24
REMC_PSDIV_16	0x0001	Clock Divide PCLK/16
REMC_PSDIV_12	0x0000	Clock Divide PCLK/12
REMC_INT_ENABLE	0	Interrupt enabled
REMC_INT_DISABLE	1	Interrupt disabled

4.10 The Compiling Option

This sample program can enable/disable unused functions using the compiling option (or Symbol definition). The functions are disabled by default value, which must be noted when compiling a program.

Symbol list

Symbol name	Description
FOR_REMC_TRANSFER	Enables codes for the output process.
FOR_REMC_RECEIVE	Enables codes for the input process.

* Enabled in the Make file (remc_gnu17IDE.mak) of the sample program.

5. NOTES

This sample program uses TV code (18E7h) proprietary to NEC Corporation.
Please prepare the custom code on your side for your own applications.

Also please note and comply with the following requirements given by NEC Corporation.

- Do not use the code in new custom codes without previous permission by NEC Corporation.
- When using existing custom codes from other companies, use them under your own responsibility with careful attention.

REVISION HISTORY

Date	revision	modification	note
08/04/30	1.0	New issue	

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
2580 Orchard Parkway
San Jose , CA 95131,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15 Muenchen Bayern,
80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg.,No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.
SHENZHEN BRANCH
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 411449500
First issue March 2008 in JAPAN Ⓗ