

S1C17801

UART Application Note

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

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

OVERVIEW

This document is the reference to use the UART 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 must be installed.

The USB driver for the SVT17 ICD board must be installed.

Table of Contents

1. SPECIFICATIONS	1
2. FUNCTIONAL DESCRIPTION	3
3. SOFTWARE DESCRIPTION	4
3.1 File Configuration	4
3.2 Module Description.....	5
3.3 Global Variables	5
3.4 Structures	6
3.5 Operation Procedure	6
3.6 Outline of sample program operations	7
3.7 Flowcharts	8
3.8 UART Driver Detailed Description	12
3.9 Header Definitions	17
REVISION HISTORY	18

1. SPECIFICATIONS

Connect cables of SIN and SOUT terminals of the two SVT17801 CPU boards or one SVT17801 CPU board.

After program execution, the system starts signal transmission and reception, and then enters the signal waiting status.

When any of SW1 to SW6 is pressed, the transmission data is written in UART transmission data registers.

[Transmission]

The 0x21 to 0x7e ASCII character codes are sent as the transmission data.

[Reception]

Up to 100 data sets are read for each reception.

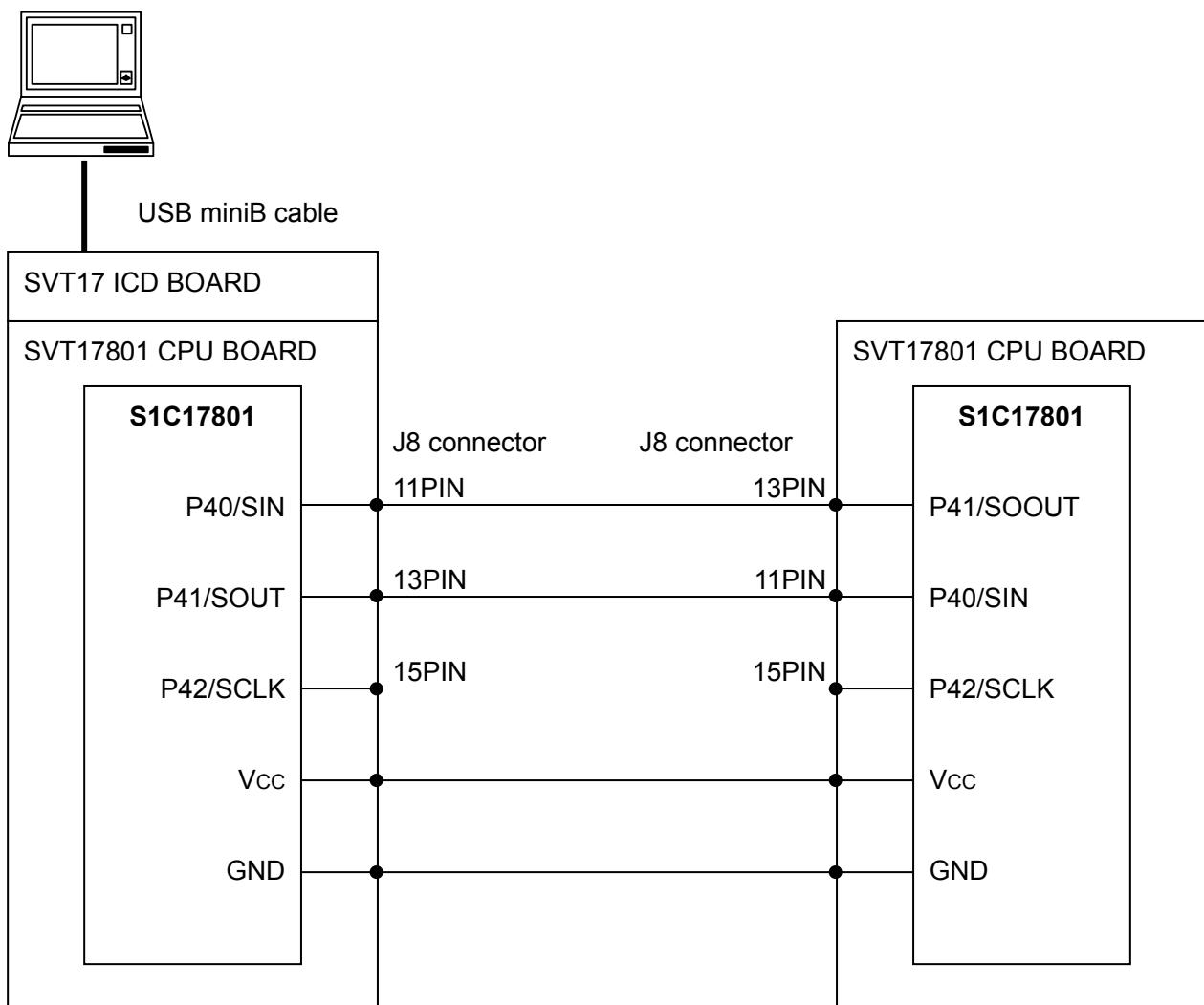
When all data is sent and when 100 data sets are received, the UART communication terminates.

All received data is displayed on the simulated I/O.

The following shows the system connection.

When connecting two SVT17801 CPU boards

PC (having the GNU17 development tool)

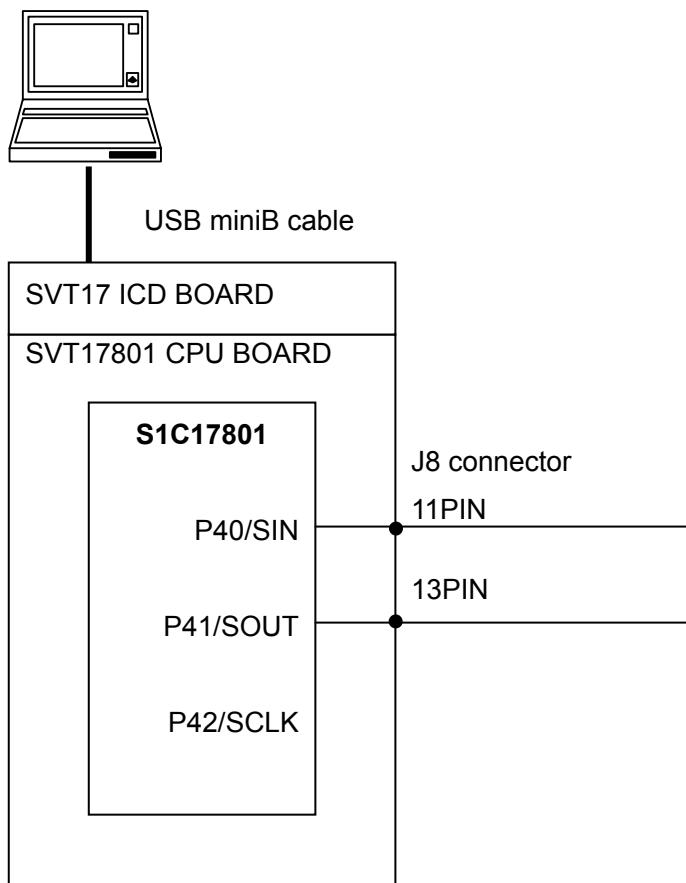


* Connect the SCLK pin only when an external clock is used as the signal transmission clock.

1. SPECIFICATIONS

When connecting one SVT17801 CPU board

PC (having the GNU17 development tool)



2. FUNCTIONAL DESCRIPTION

I/O pins The UART I/O pins (SIN, SOUT, and SCLK) are also used as general I/O port pins (P40, P41, and P42).

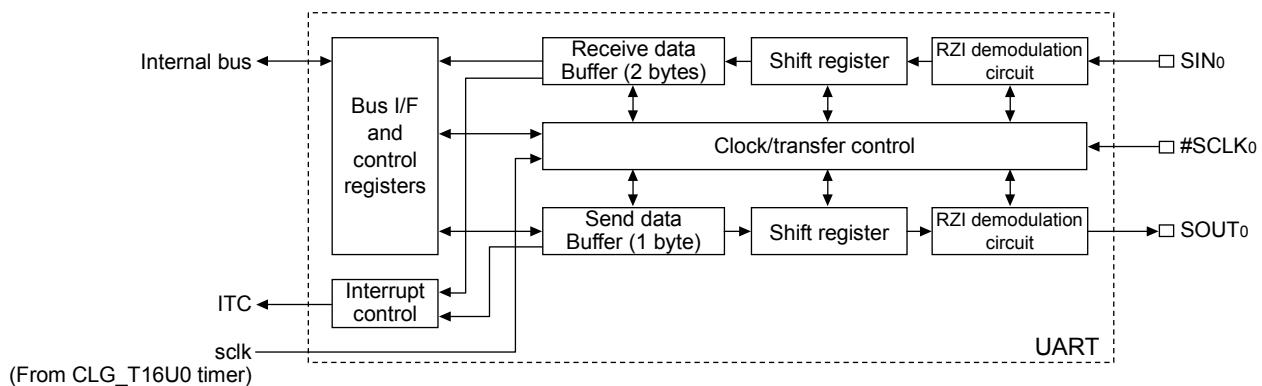
Transfer clock The internal clock or an external clock can be selected as the UART signal transmission clock.

Interrupt The vector number and address of UART interrupt are as follows:

Vector number: 16(0x10)

Vector address: 0x20040

Three types of interrupts such as a send buffer empty interrupt, a receive buffer full interrupt, and a receive error interrupt can be generated.



3. SOFTWARE DESCRIPTION

3. SOFTWARE DESCRIPTION

3.1 File Configuration

File name	Function
boot.c	Startup module
main.c	Main function
inthdlr.c	Interrupt handler function
lib.c	Library settings
vector.c	Vector table settings
header/reg_801.h	Register definitions
header/vector.h	Vector table definitions
uart_drv/uart.c	UART driver API group
uart_drv/uart_api.h	UART driver API definitions
uart_drv/uart.h	UART driver header definitions
uart_mdl/uartm_mdl.c	UART module API group
uart_mdl/uartm_api.h	UART module API definitions
clg_drv/	CLG driver group folder
cmu_drv/	CMU driver group folder
flashc_drv/	FLASHC driver group folder
gpio_drv/	GPIO driver group folder
itc_drv/	ITC driver group folder
sramc_drv/	SRAMC driver group folder
uart_gnu17IDE.lds	Linker script file
uart_gnu17IDE.cmd	GDB command file
uart_gnu17IDE.par	Parameter configuration file
uart_gnu17IDE.mak	Make file
.cdtproject	Project file
.gnu17project	Project file
.project	Project file
GDB17 Launch for uart.launch	Project startup file

3.2 Module Description

File name: main.c

Function name	Function
main	Calls the ITC/PORT initialization function, sets an interrupt, and calls the driver sample subfunction (uartDriverSample).
sendString	Sends the transmission data.
UartIntHandler	UART interrupt function Determines an interrupt type. When the send data buffer is empty: This function sends the transmission data if it is not NULL. When the receive data buffer is full: This function receives data if it is 100 data sets or less. Clears the interrupt flag.
uartDriverSample	This is a subfunction that uses the UART driver API. This function sets the clg, sets the UART, controls data transmission, and displays them on the simulated I/O.
PortInitialize	The port setup function Sets the port to be used for the UART.
ItcInitialize	Initializes the interrupt controller. This function disables all interrupts.
ClgInitialize	The CLG operation setup function Starts the CLG_T16U operation with the following settings: Baud rate at 115200 bps, PCLK 1/1, and in REPEAT counter mode
uartModuleSample	This is a subfunction that uses the UART module API. Initializes the UART module, controls data transmission, and displays them on the simulated I/O.

3.3 Global Variables

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

Variable name	Type	Function
strCount	unsigned int	Stores the sent data count.
receiveCount	unsigned int	Stores the received data count.
portFlag	unsigned char	Stores the P0-port input interrupt bits.
*strData	unsigned char	Specifies the send data address
receiveData[110]	unsigned char	Stores the received data.

3. SOFTWARE DESCRIPTION

3.4 Structures

The following lists the structures used in the sample program.

Definition name		
T_UART_CFG config		
Members		
clock	unsigned char	Sets the clock (0 for internal clock, 1 for external clock)
stop	unsigned char	Sets the stop bits (0 for 1 bit, 1 for 2 bits)
parityMode	unsigned char	Sets a parity mode (even or odd)
parityEnable	unsigned char	Sets a parity (no parity or with parity)
length	unsigned char	Sets a data length (7 bits or 8 bits)
txIntEnable	unsigned char	Enables or disables a data send interrupt
rxIntEnable	unsigned char	Enables or disables a receive data interrupt
errIntEnable	unsigned char	Enables or disables a receive error interrupt
rxIntTiming	unsigned char	Sets a receive data interrupt trigger (1-byte or 2-byte data reception)
intFunc	void *	Interrupt function pointer
Remarks		
UART initialization structures.		

3.5 Operation Procedure

Importing the project

- (1) Start the IDE, and import the “uart” project.

* For the import procedure, refer to Section 3 “Software Development Procedure” of the S5U1C17001C Manual.

Build

- (1) Build the “uart” project using the IDE.

Cabling and Power-On

- (1) Connect the SVT17801, USB miniB port and PC using their cables.

- (2) Connect the UART pins of two SVT17801 CPU boards or one SVT17801 CPU board.

SIN (J8 11, TP42)

SOUT (J8 13, TP43)

SCLK (J8 15, TP44) * Connect the SCLK pin only when an external clock is used for data transmission.

- (3) Reset the SVT17 ICD board.

Execution

- (1) Execute the “uart” project using the IDE.

- (2) After “uart” execution, press either of SW1 to SW6 by following the instruction displayed on the simulated I/O.

3.6 Outline of sample program operations

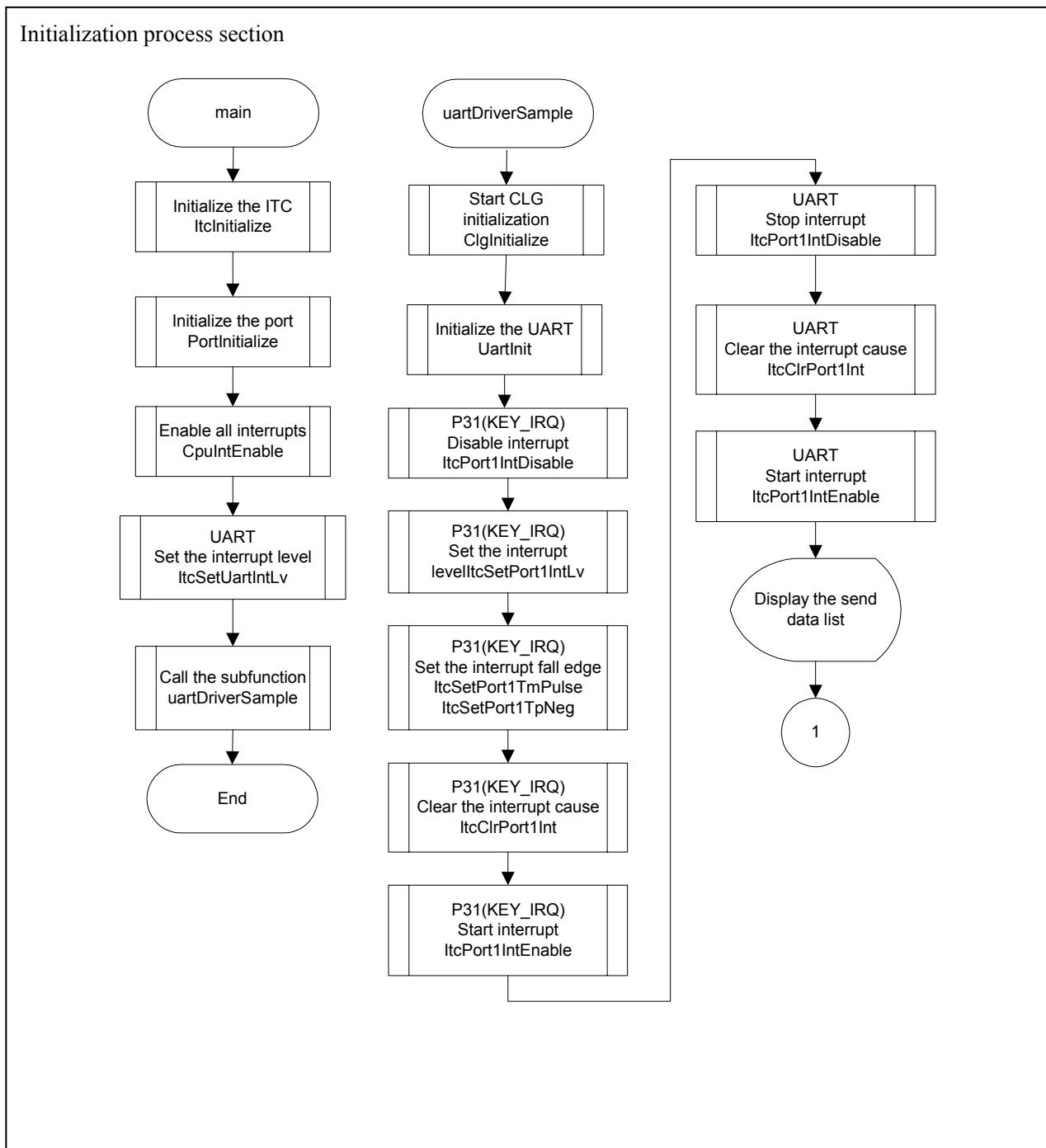
- (1) The CLG_T16U is initialized. (The transfer rate is set to 115200 bps.)
- (2) The UART is initialized.
- (3) The UART interrupts (such as the send buffer empty interrupt and the receive buffer full interrupt) are initialized.
- (4) The UART is started and an UART interrupt in the ITC is enabled.
 - * When any of SW1 to SW6 is pressed in the receiving state, the data transmission starts.
- (5) Data is sent.
 - * The interrupt function continues to send data until the send data buffer is emptied.
- (6) Data is received.
 - *: The interrupt function continues to read the receive data until 100 data sets are received.
- (7) After reception of 100 data sets, an UART interrupt in the ITC is disabled and the CLG_T16U and UART are stopped.
- (8) The received data is displayed on the simulated I/O.

* During interrupt waiting, the system enters the Halt mode to save the electric power.

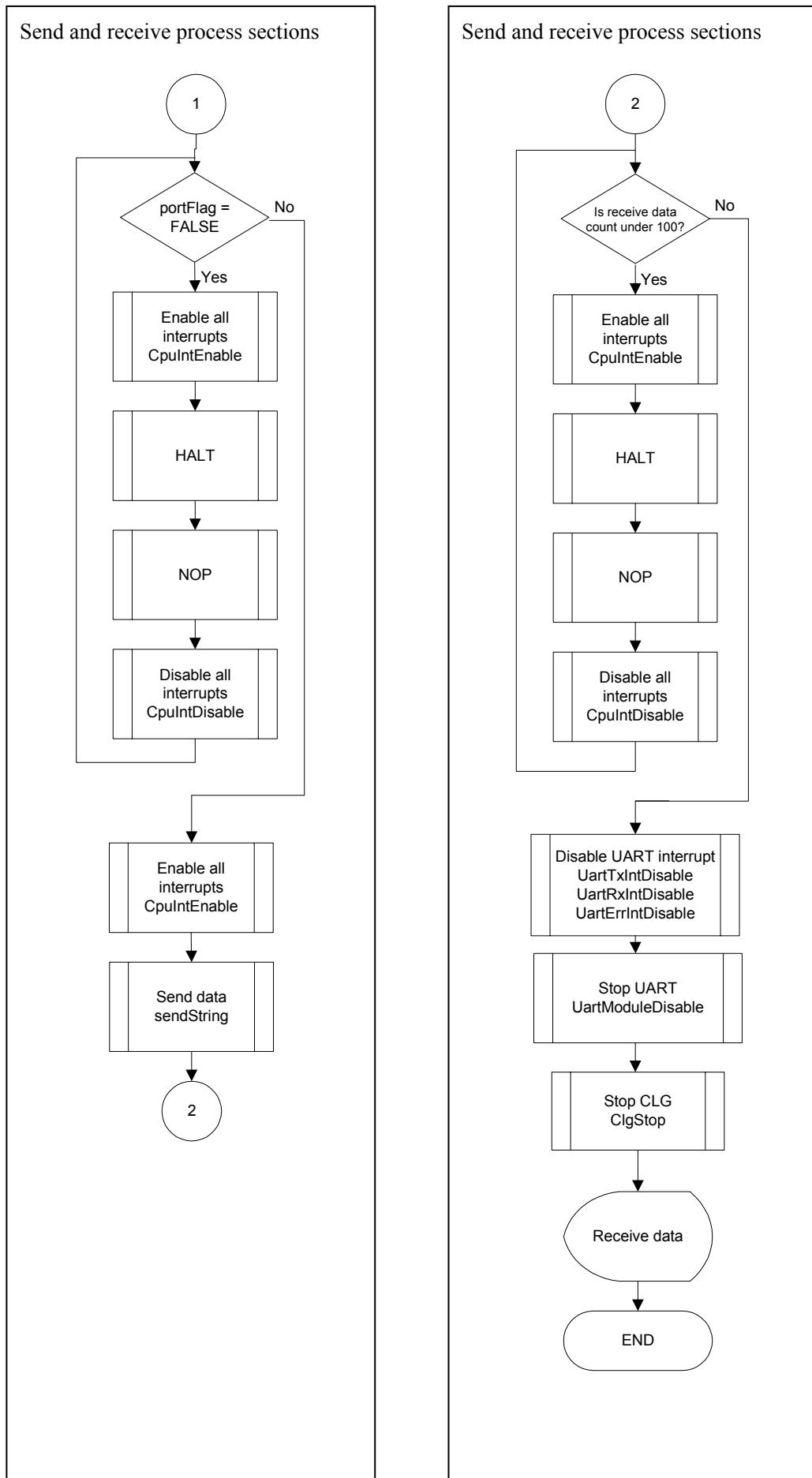
3. SOFTWARE DESCRIPTION

3.7 Flowcharts

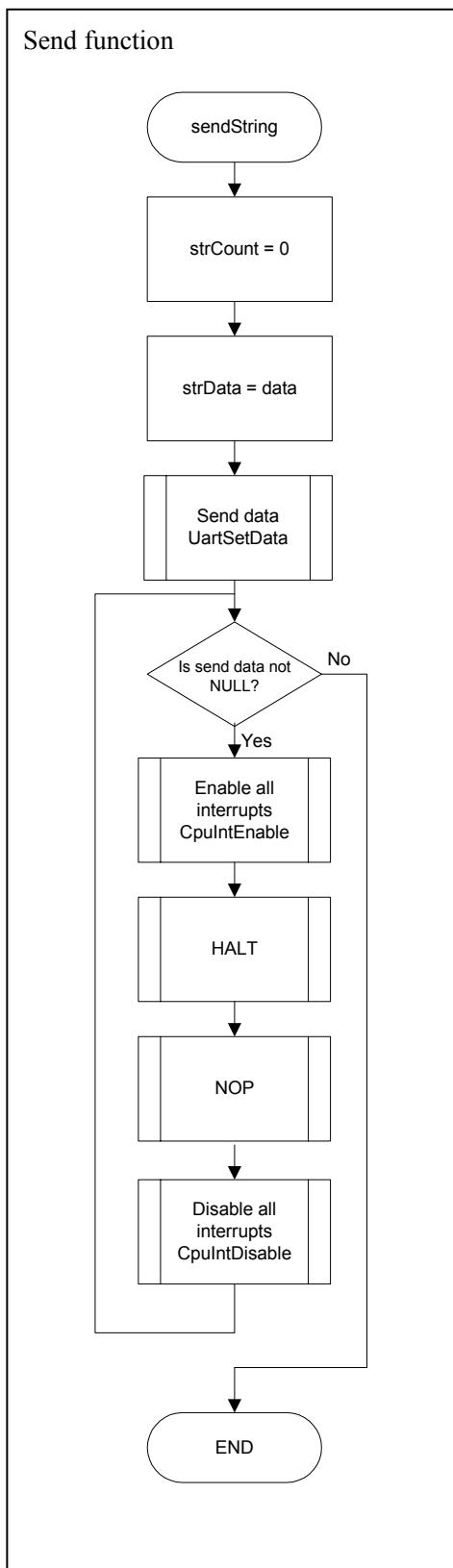
The following shows flowcharts of the main routine and interrupt functions.

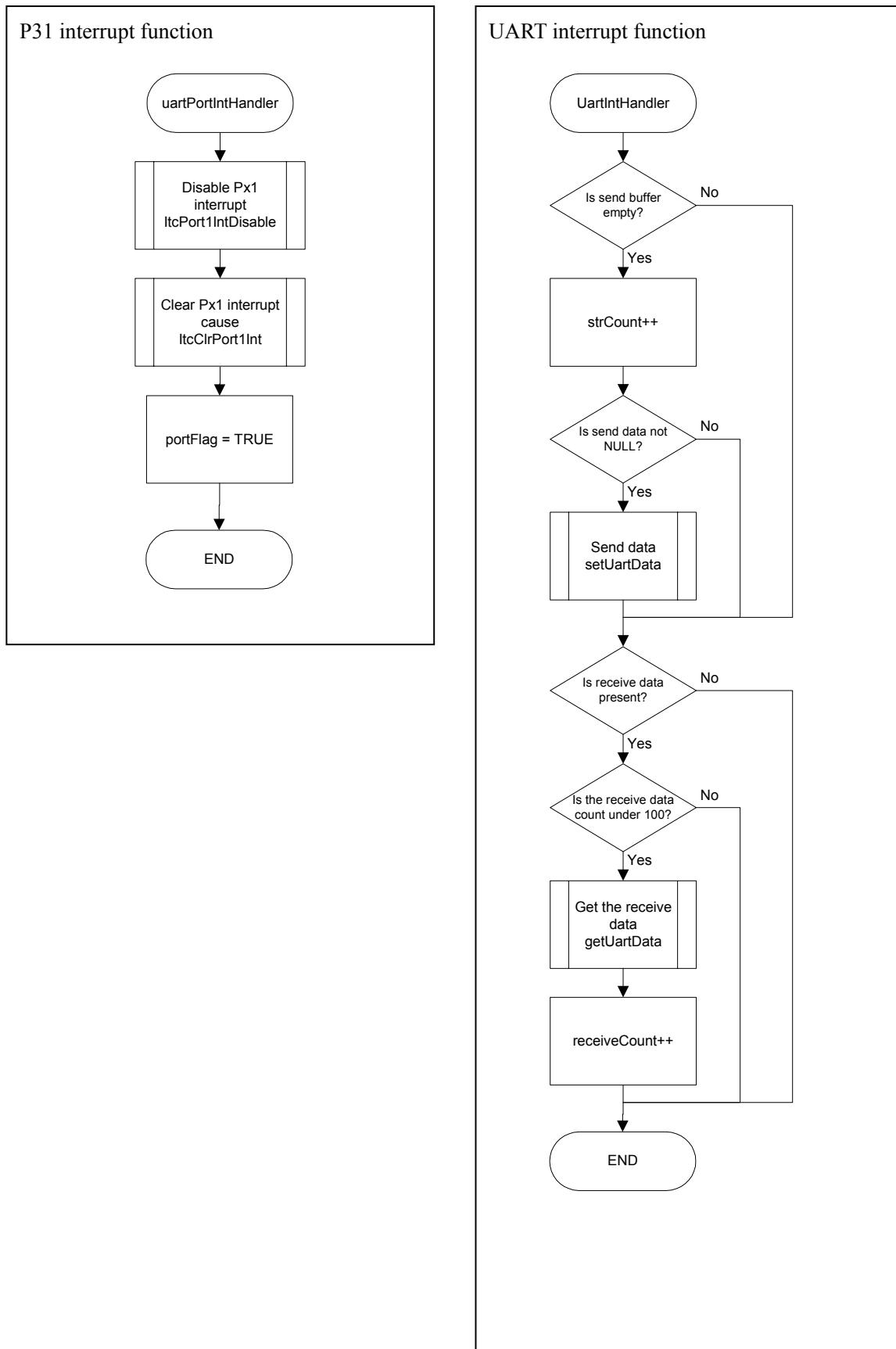


3. SOFTWARE DESCRIPTION



3. SOFTWARE DESCRIPTION





3. SOFTWARE DESCRIPTION

3.8 UART Driver Detailed Description

The following lists the functions described in uart.c and uart.h.

UART Initialization

Format	void UartInit(T_UART_CFG *pConfig)
Function	Sets the UART controller
Arguments	pConfig -in Communication setup parameter
Return values	None
(Process description)	
(1) The UART is disabled (by call of UartModuleDisable). (2) The input clock, stop bit length, parity, and data length are set. (3) The receive buffer full trigger, send data interrupt, receive data interrupt, receive buffer full trigger, data send interrupt, data receive interrupt, and data receive error interrupt are set. (4) The IrDA mode is turned OFF. (5) The interrupt function is registered if it is set. (6) The UART is started (by call of UartModuleEnable).	
Initialization parameter structure:	
typedef struct{ unsigned char clock; // input clock unsigned char stop; // stop bit unsigned char parityMode; // parity mode select unsigned char parityEnable; // parity enable unsigned char length; // character length unsigned char txIntEnable; // tx interrupt unsigned char rxIntEnable; // rx interrupt unsigned char errIntEnable; // rx err interrupt unsigned char rxIntTiming; // receive buffer full interrupt condition void (*intFunc)(void); }T_UART_CFG;	

UART Interrupt Handler

Format	void UartIntProc(void)
Function	Calls the UART interrupt process.
Arguments	None
Return values	None
(Process description) (1) The specified interrupt process is called.	

Send Interrupt Enable (Macro)

Format	void UartTxIntEnable(void)
Function	Enables a send data interrupt.
Arguments	None
Return values	None
(Process description) (1) The Send Interrupt Enable bit is set to logical 1.	

Send Interrupt Disable (Macro)

Format	void UartTxIntDisable(void)
Function	Disables the send data interrupt.
Arguments	None
Return values	None
(Process description)	(1) The Send Interrupt Enable bit is cleared.

Send Interrupt Status Check (Macro)

Format	char UartGetTxInt(void)
Function	Checks the send interrupt status.
Arguments	None
Return values	Indicates the send data interrupt status. TRUE Interrupt enabled FALSE Interrupt disabled
(Process description)	(1) The Send Interrupt Enable bit value is obtained.

Receive Interrupt Enable (Macro)

Format	void UartRxIntEnable(void)
Function	Enables the receive data interrupt.
Arguments	None
Return values	None
(Process description)	(1) The Receive Interrupt Enable bit is set to logical 1.

Receive Interrupt Disable (Macro)

Format	void UartRxIntDisable(void)
Function	Disables the receive data interrupt.
Arguments	None
Return values	None
(Process description)	(1) The Receive Interrupt Enable bit is cleared.

Receive Interrupt Status Check (Macro)

Format	char UartGetRxInt(void)
Function	Checks the receive interrupt status.
Arguments	None
Return values	Indicates the receive data interrupt status. TRUE Interrupt enabled FALSE Interrupt disabled
(Process description)	(1) The Receive Interrupt Enable bit value is obtained.

3. SOFTWARE DESCRIPTION

Receive Error Interrupt Enable (Macro)

Format	void UartErrIntEnable(void)
Function	Enables a receive error interrupt.
Arguments	None
Return values	None
(Process description) (1) The Receive Error Interrupt Enable bit is set to logical 1.	

Receive Error Interrupt Disable (Macro)

Format	void UartErrIntDisable(void)
Function	Disables a receive error interrupt.
Arguments	None
Return values	None
(Process description) (1) The Receive Error Interrupt Enable bit is cleared.	

Receive Error Interrupt Status Check (Macro)

Format	char UartGetErrInt(void)
Function	Checks the receive error interrupt status.
Arguments	None
Return values	Indicates the receive error interrupt status. TRUE Interrupt enabled FALSE Interrupt disabled
(Process description) (1) The Send Error Interrupt Enable bit value is obtained.	

Send Buffer Status Check (Macro)

Format	char UartCheckTxBuffer(void)
Function	Checks the send data buffer status.
Arguments	None
Return values	Indicates the send data buffer status. UART_TX_BUFF_EMPTY Buffer empty UART_TX_BUFF_FULL Buffer full
(Process description) (1) The Send Buffer Empty Flag bit value is obtained.	

Receive Buffer Status Check (Macro)

Format	char UartCheckRxBuffer(void)
Function	Checks the receive data buffer status.
Arguments	None
Return values	Indicates the receive data buffer status. UART_RX_BUFF_FULL Buffer full UART_RX_BUFF_EMPTY Buffer empty
(Process description) (1) The Receive Data Ready flag bit value is obtained.	

Receive Error Interrupt Cause Check (Macro)

Format	char UartCheckErrIntFactor(void)
Function	Checks the Receive Error flag status.
Arguments	None
Return values	Indicates the receive error status. TRUE A receive error occurred. FALSE No receive error occurred.
(Process description) (1) The framing error bits, parity error bits, and overrun error bits are obtained, and they are logically adds (ORed).	

Send Busy Status Check (Macro)

Format	char UartCheckTrnsBusy(void)
Function	Checks the send data busy status.
Arguments	None
Return values	Indicates the receive error status. UART_TRNS_BUSY Busy status UART_TRNS_IDLE Transfer idling status
(Process description) (1) The Transfer Busy flag bit value is obtained.	

Receive Error Interrupt Cause Clear (Macro)

Format	void UartClearErrIntFactor(void)
Function	Clears the Receive Error Interrupt flag.
Arguments	None
Return values	None
(Process description) (1) The framing error bit, parity error bit, and overrun error bit are set to logical 1.	

Data Send (Macro)

Format	void UartSetData(char data)
Function	Clears the Receive Error Interrupt flag.
Arguments	data -in Send data
Return values	None
(Process description) (1) The specified data is written in registers.	

Data Receive (Macro)

Format	char UartGetData(void)
Function	Clears the Receive Error Interrupt flag.
Arguments	None
Return values	Indicates the received data.
(Process description) (1) The received data is read.	

3. SOFTWARE DESCRIPTION

UART Module Enable (Macro)

Format	void UartModuleEnable(void)
Function	Starts the UART module.
Arguments	None
Return values	None
(Process description)	(1) The UART module enable bit is set to logical 1.

UART Module Disable (Macro)

Format	void UartModuleDisable(void)
Function	Stops the UART module.
Arguments	None
Return values	None
(Process description)	(1) The UART module enable bit is cleared.

3.9 Header Definitions

The following lists the header definitions used in the driver functions.

Definition name	Value	Description
UART_BAUDRATE_115200	0x0019	Sets the baud rate to 115200 bps (PCLK=48 MHz).
UART_BAUDRATE_57600	0x0033	Sets the baud rate to 57600 bps (PCLK=48 MHz).
UART_BAUDRATE_38400	0x004D	Sets the baud rate to 38400 bps (PCLK=48 MHz).
UART_BAUDRATE_19200	0x009B	Sets the baud rate to 19200 bps (PCLK=48 MHz).
UART_BAUDRATE_14400	0x00CF	Sets the baud rate to 14400 bps (PCLK=48 MHz).
UART_BAUDRATE_9600	0x0137	Sets the baud rate to 9600 bps (PCLK=48 MHz).
UART_BAUDRATE_4800	0x0270	Sets the baud rate to 4800 bps (PCLK=48 MHz).
UART_BAUDRATE_2400	0x04E1	Sets the baud rate to 2400 bps (PCLK=48 MHz).
UART_BAUDRATE_1200	0x09C3	Sets the baud rate to 1200 bps (PCLK=48 MHz).
UART_INTERNAL_CLOCK	0	Internal clock
UART_EXTERNAL_CLOCK	1	External clock
UART_1STOP_BIT	0	Sets one (1) stop bit.
UART_2STOP_BIT	1	Sets two (2) stop bits.
UART_EVEN_PARITY	0	Even parity
UART_ODD_PARITY	1	Odd parity
UART_NO_PARITY	0	Disables the parity bits.
UART_WITH_PARITY	1	Enables the parity bits.
UART_7BIT_LENGTH	0	Sets the character length to 7 bits.
UART_8BIT_LENGTH	1	Sets the character length to 8 bits.
UART_INT_DISABLE	0	Disables an interrupt.
UART_INT_ENABLE	1	Enables an interrupt.
UART_RX_INT_1BYTE_RECV	0	Sets the receive buffer full interrupt condition to 1 byte.
UART_RX_INT_2BYTE_RECV	1	Sets the receive buffer full interrupt condition to 2 bytes.
UART_TX_BUFF_EMPTY	1	Sets the Send Buffer Empty flag to the buffer full status.
UART_TX_BUFF_FULL	0	Sets the Send Buffer Empty flag to the data presence status.
UART_RX_BUFF_EMPTY	0	Sets the Receive Buffer Ready flag to the buffer empty status.
UART_RX_BUFF_FULL	1	Sets the Receive Buffer Ready flag to the data presence status.
UART_TRNS_BUSY	1	Sets the Transfer Busy flag to the busy status.
UART_TRNS_IDLE	0	Sets the Transfer Busy flag to the idle status.

REVISION HISTORY

REVISION HISTORY

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