SH7216 Group

Using the LCD Touchscreen

R01AN0192EJ0100
Rev. 1.00
Nov. 29, 2010

Summary

This application note describes an example to control the touchscreen using the SH7216 A/D Converter and general-purpose I/O ports.

Target Device

SH7216 MCU

Contents

1. Introduction

1.1 Specifications

Inputs and calculates the voltage of the pen-down point on the touchscreen to the A/D Converter to convert the voltage into x-/y-coordinate data.

1.2 Modules Used

- A/D Converter
- General-purpose I/O ports

1.3 Applicable Conditions

MCU	SH7216
Operating Frequency	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated Development Environment	Renesas Electronics Corporation High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ compiler package Ver.9.03 Release 00
Compiler Options	-cpu=sh2afpu -fpu=single -include="$(WORKSPDIR)\inc" -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo

1.4 Related Application Notes

For more information, refer to the following application note:

- SH7216 Group Example of Initialization

2. Applications

The SH7216 A/D Converter detects the voltage of the pen-down point on the touchscreen, and calculates the x-/y-coordinates from the A/D value.

2.1 Touchscreen

Table 1 lists the specifications of the LCD touchscreen used in this application.

Table 1 LCD Touchscreen Specifications

Item		Description
Touchscreen type		Resistive touchscreen
Resolution		QVGA
Number of pixels		H 240 × V 320 (Number of dots: H (240 × 3) × V 320)
Resistance between terminals	Top-Bottom (Y-axis)	250 to 500 Ω
	Left-Right (X-axis)	200 to 650 Ω
Linearity	Top-Bottom (Y-axis)	± 1.5% (max.)
	Left-Right (X-axis)	± 1.5% (max.)
Applied voltage		5 VDC

A resistive touchscreen is provided voltages either at x-axis or y-axis, and the pen-down point is detected from the other axis (See Figure 1). This application calculates the A/D average value between Top and Bottom as the x-coordinate voltage (Left to Right), and calculates the A/D average value between Left and Right as the y-coordinate voltage (Top to Bottom).

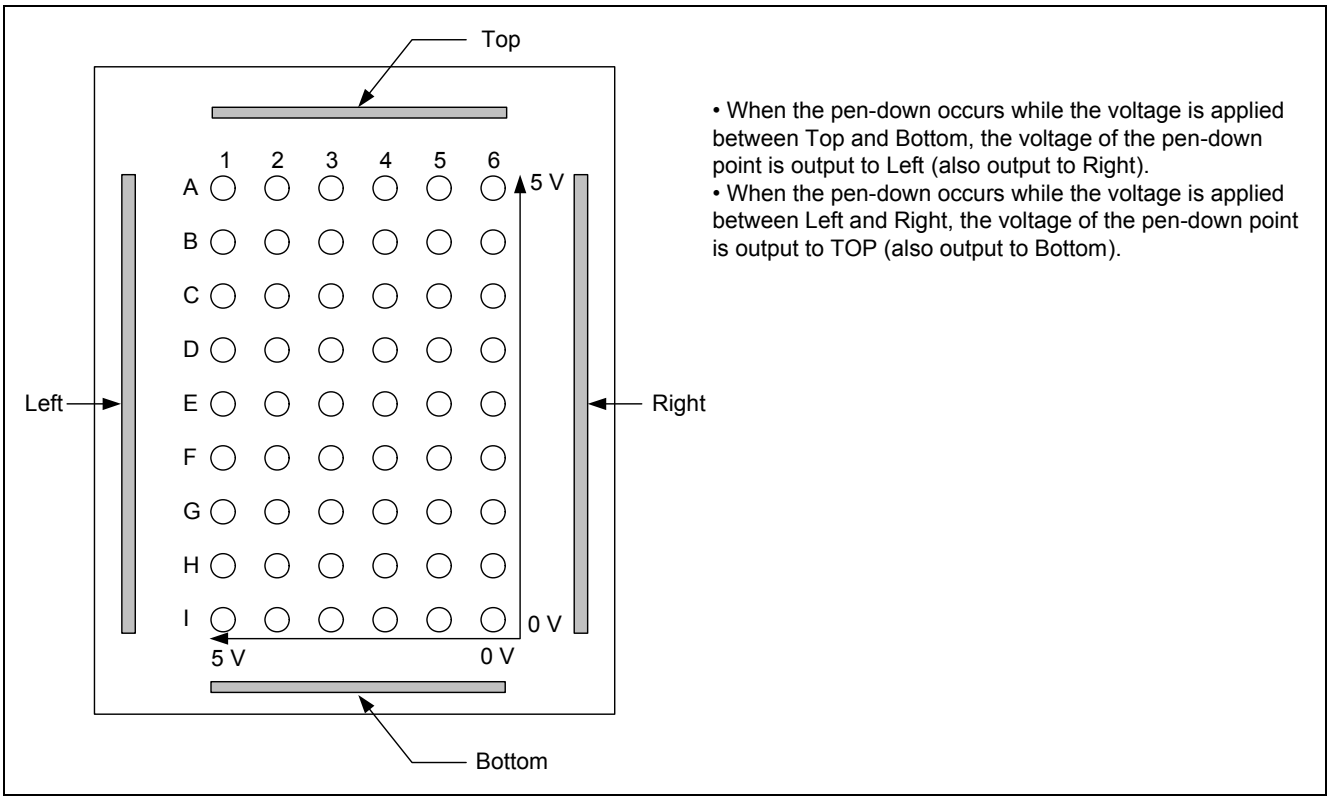

Figure 1 Detecting the Voltage of the Pen-down Point on Touchscreen

2.2 Circuit Configuration

This section describes the touchscreen circuit example. The SH7216 ports (PE0 to PE3) turn ON or OFF the transistor to apply the voltage to Top, Bottom, Left, and Right on the touchscreen, and the port (PE7) applies the voltage to detect the pen-down point by IRQ4. Outputs from the touchscreen are inputs to the SH7216 A/D Converter. Figure 2 shows the touchscreen circuit example.

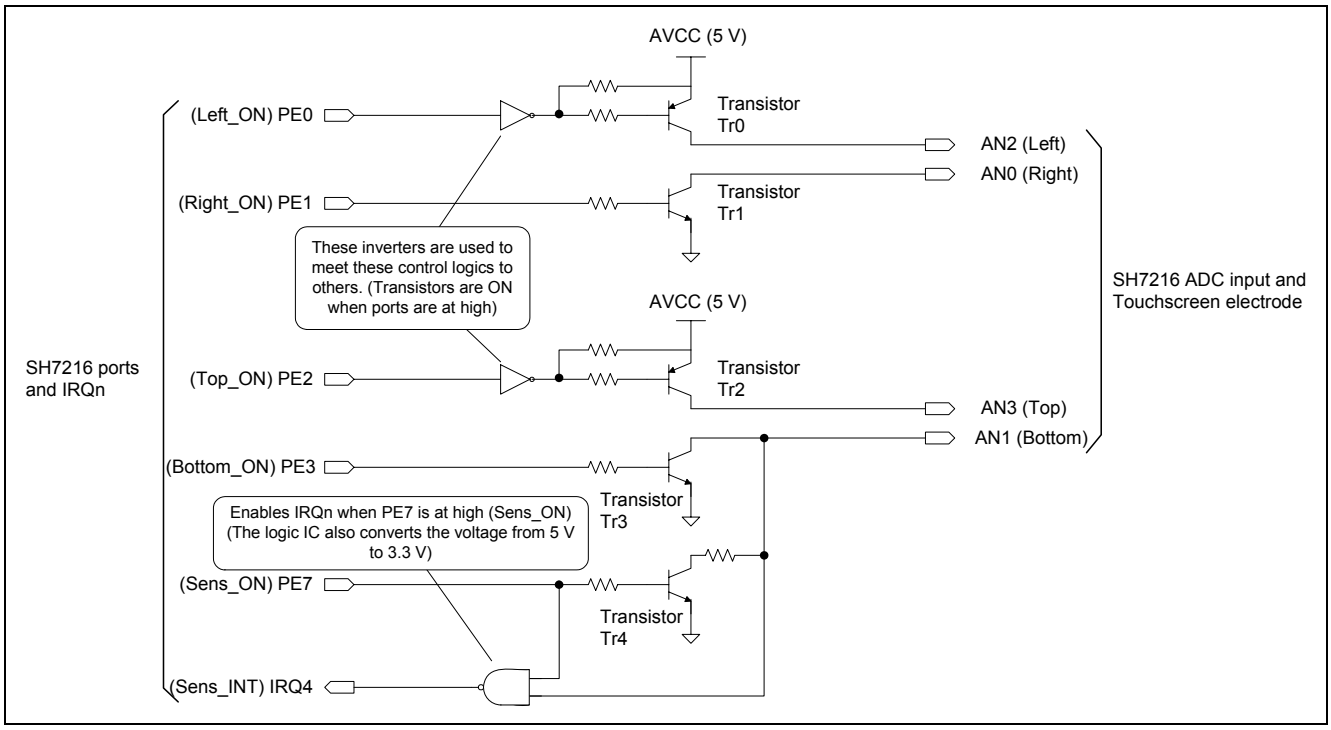

Figure 2 Touchscreen Circuit

2.3 Sample Program Operation

This section describes the sample program operation.

2.3.1 Detecting the Pen-down Point

Detecting the pen-down:

Set ports PE7 and PE0 to high level to turn ON the transistors Tr0 and Tr4, or set other ports to low level to turn OFF the transistors Tr1 to Tr3.
IRQ4 interrupt occurs when the pen-down occurs.

Detecting y-coordinate:

Set ports PE2 and PE3 to high level to turn ON the transistors Tr2 and Tr3, or set other ports to low level to turn OFF transistors Tr0, Tr1, and Tr4.
Y-coordinate voltage of the pen-down point is output to Left and Right, and A/D Converter AN0 and AN2 detects the voltage level.

Detecting x-coordinate:

Set ports PE0 and PE1 to high level to turn ON the transistors Tr0 and Tr1, or set other ports to low level to turn OFF the transistors Tr2 to Tr4.
X-coordinate voltage of the pen-down point is output to Top and Bottom, and A/D Converter AN1 and AN3 detects the voltage level.

2.3.2 Calculating Coordinates

Figure 3 shows the flow chart for detecting the voltage of the pen-down point and calculating the coordinates.

Limit, offset, and coefficients (the ratio of the number of pixels and valid A/D range) in the flow chart below depends on the type of touchscreen. Measure these values with your touchscreen to decide the values. This application repeats measuring the A/D value when pen-down occurs at four corners of the H: 240 × V: 320 touchscreen to decide the limit and offset values.

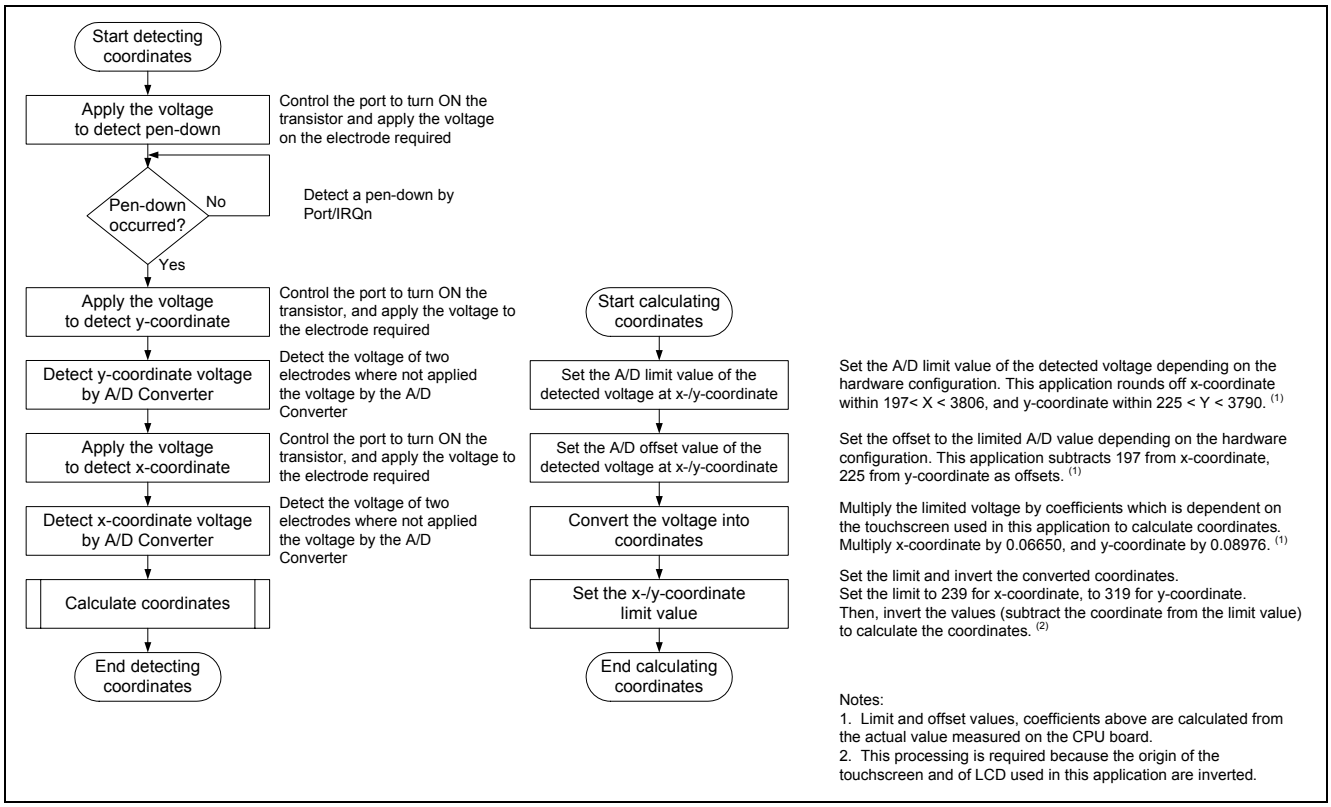

Figure 3 Flow Chart for Detecting the Voltage of the Pen-down Point and Calculating Coordinates

2.4 Sample Program Procedure

This section describes an example to set the SH7216 peripheral functions. Table 2 lists the register settings for the A/D Converter. Figure 4 shows the flow chart of the sample program.

Table 2 A/D Converter Register Setting

Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'7A	• MSTP32 = "0": Supplies the clock to A/D_0
A/D control register_0 (ADCR_0)	H'FFFF E800	H'90	• ADST = "1": Starts the A/D conversion • ADCS = "0": Specifies single-cycle scan mode • ACE = "0": Disables to clear the ADDR register automatically by reading the ADDR register • ADIE = "1": Enables to generate the A/D conversion end interrupt • TRGE = "0": Disables the A/D conversion by the external trigger input or the A/D converter start trigger from the Multi-function Timer Pulse Unit 2 • EXTRG = "0" Activates the A/D Converter by the A/D converter start trigger from the Multi-function Timer Pulse Unit 2
A/D analog input channel select register_0 (ADANSR_0)	H'FFFF E820	H'01	• ANS0 = "1": Specifies AN0
		H'02	• ANS1 = "2": Specifies AN1
		H'04	• ANS1 = "4": Specifies AN2
		H'08	• ANS1 = "8": Specifies AN3

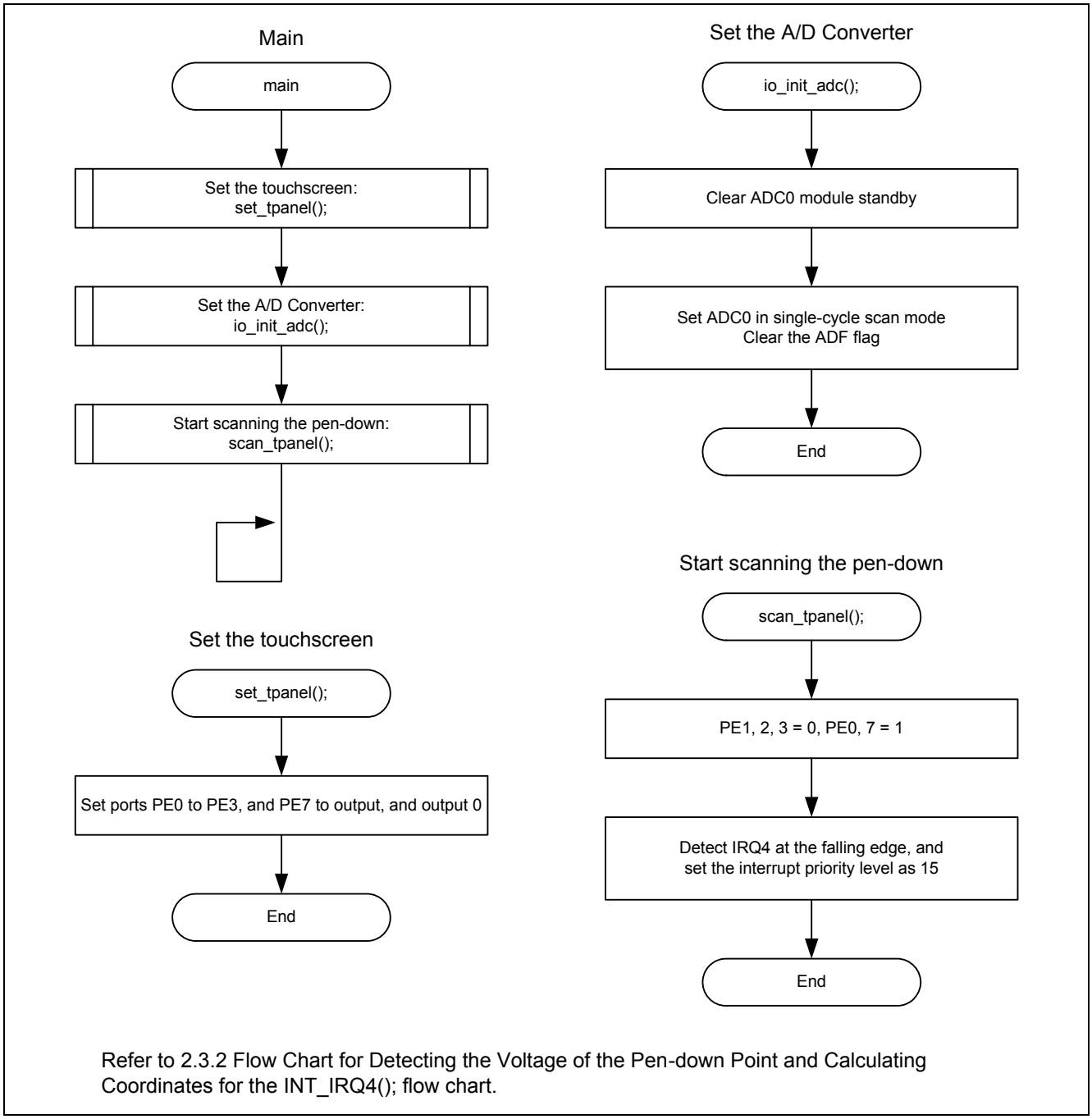

Figure 4 Sample Program Flow Chart

3. Sample Program Listing

3.1 Sample Program Listing "main.c" (1/9)

```
1    /***********************************************************************
2    *   DISCLAIMER
3    *
4    *   This software is supplied by Renesas Electronics Corp. and is only
5    *   intended for use with Renesas products.  No other uses are authorized.
6    *
7    *   This software is owned by Renesas Electronics Corp. and is protected under
8    *   all applicable laws, including copyright laws.
9    *
10   *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11   *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12   *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13   *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14   *   DISCLAIMED.
15   *
16   *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17   *   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18   *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19   *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20   *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21   *
22   *   Renesas reserves the right, without notice, to make changes to this
23   *   software and to discontinue the availability of this software.
24   *   By using this software, you agree to the additional terms and
25   *   conditions found by accessing the following link:
26   *   http://www.renesas.com/disclaimer
27   ************************************************************************
28   *   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
29   ************************************************************************/
30   /*""FILE COMMENT""************ Technical reference data ********************
31   *   System Name : SH7216 Sample Program
32   *   File Name   : main.c
33   *   Abstract    : Touch panel usage example Application
34   *   Version     : 1.00.00
35   *   Device      : SH7216
36   *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
37   *               : C/C++ compiler package for the SuperH RISC engine family
38   *               :                             (Ver.9.03 Release00).
39   *   OS          : None
40   *   H/W Platform: R0K572167 (CPU board)
41   *   Description :
42   ************************************************************************
43   *   History     : Sep.14,2010 Ver.1.00.00
44   *""FILE COMMENT END""*********************************************************/
45   #include <machine.h>
46   #include "iodefine.h"
47   #include "vect.h"
48
```

3.2 Sample Program Listing "main.c" (2/9)

```
49   /* ==== Prototype declaration ==== */
50   void main(void);
51   void set_tpanel(void);
52   void io_init_adc(void);
53   void scan_tpanel(void);
54   void port_wait(int time);
55   void detect_tpanel(void);
56
57   int pen_down;
58
59   /* ==== RAM allocation variable declaration ==== */
60   unsigned short Y_adr;         /* Y-coordinate */
61   unsigned short X_adr;         /* X-coordinate */
62
63   /*""FUNC COMMENT""*****************************************************************
64    * ID           :
65    * Outline      : Sample program main
66    *-------------------------------------------------------------------------------
67    * Include      :
68    *-------------------------------------------------------------------------------
69    * Declaration  : void main(void);
70    *-------------------------------------------------------------------------------
71    * Description  : Sample program main
72    *-------------------------------------------------------------------------------
73    * Argument     : void
74    *-------------------------------------------------------------------------------
75    * Return Value : void
76    *-------------------------------------------------------------------------------
77    * Note         : None
78    *""FUNC COMMENT END""*************************************************************/
79   void main(void)
80   {
81     /* ==== Sets the touchscreen ==== */
82     set_tpanel();
83
84     /* ==== Sets the ADC ==== */
85     io_init_adc();
86
87     /* ==== Starts scanning the pen-down ==== */
88     scan_tpanel();
89
90     while(1){
91       if(pen_down==1){
92           detect_tpanel();
93           pen_down==0;
94       }
95     }
96   }
97
```

3.3 Sample Program Listing "main.c" (3/9)

```
98    /*""FUNC COMMENT""*************************************************************
99     * ID           :
100    * Outline      : Port setting to control the touchscreen
101    *-------------------------------------------------------------------------------
102    * Include      : #include "iodefine.h"
103    *-------------------------------------------------------------------------------
104    * Declaration  : void set_tpanel(void)
105    *-------------------------------------------------------------------------------
106    * Function     : Sets ports PE0, PE1, PE2, PE3, and PE7 as output port,
107    *              : and to output 0.
108    *-------------------------------------------------------------------------------
109    * Argument     : void
110    *-------------------------------------------------------------------------------
111    * Return value : void
112    *-------------------------------------------------------------------------------
113    * Notice       : none
114    *""FUNC COMMENT END""*********************************************************/
115   void set_tpanel(void)
116   {
117     /* ==== Sets the PFC ==== */
118     /* ---- Sets ports ---- */
119     PFC.PECRL2.BIT.PE7MD = 0;
120     PFC.PECRL1.BIT.PE3MD = 0;
121     PFC.PECRL1.BIT.PE2MD = 0;
122     PFC.PECRL1.BIT.PE1MD = 0;
123     PFC.PECRL1.BIT.PE0MD = 0;
124
125     /* ---- Sets the output value ---- */
126     PE.DR.BYTE.L = 0;
127
128     /* ---- Sets ports to output ---- */
129     PFC.PEIORL.BIT.B7 = 1;
130     PFC.PEIORL.BIT.B3 = 1;
131     PFC.PEIORL.BIT.B2 = 1;
132     PFC.PEIORL.BIT.B1 = 1;
133     PFC.PEIORL.BIT.B0 = 1;
134   }
135
```

3.4 Sample Program Listing "main.c" (4/9)

```
/*""FUNC COMMENT""*************************************************************
 * ID           :
 * Outline      : ADC configuration
 *-----------------------------------------------------------------------------
 * Include      : #include "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration  : void io_init_adc(void)
 *-----------------------------------------------------------------------------
 * Function     : Configures ADC0.
 *              : Sets ch0 to ch3 in a single-cycle scan mode.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return value : void
 *-----------------------------------------------------------------------------
 * Notice       : none
 *""FUNC COMMENT END""*********************************************************/
void io_init_adc(void)
{
  /* ==== Module standby clear ==== */
  STB.CR3.BIT._ADC0 = 0;          /* Supplies the clock to ADC0 */

  /* ==== Sets the ADC ==== */
  /* ---- Sets the A/D control register (ADCR) ---- */
  ADC0.ADCR.BYTE = 0x00;
    /*
        bit 7: ADST = 0 ---- Starts A/D conversion
        bit 6: ADCS = 0 ---- Single-cycle scan
        bit 5: ACE = 0 ----- Disables to auto-clear the ADDR by reading the ADDR
        bit 4: ADIE = 0 ---- Enables the A/D conversion end interrupt (ADI)
        bits 3, 2: Reserved(0)
        bit 1: TRGE = 0 ---- Disables the A/D conversion by the external trigger or
                            the A/D converter start trigger from MTU2/MTU2S
        bit 0: EXTRG = 0 --- Activates the A/D Converter by the A/D converter
             :               start trigger from MTU2/MTU2S
    */

  /* ---- Sets the A/D status register (ADSR) ---- */
  ADC0.ADSR.BIT.ADF = 0;          /* ADF clear */
}
```

3.5 Sample Program Listing "main.c" (5/9)

```
177  /*""FUNC COMMENT""*****************************************************************
178   * ID           :
179   * Outline      : Touchscreen scan start
180   *-------------------------------------------------------------------------------
181   * Include      : #include "iodefine.h"
182   *-------------------------------------------------------------------------------
183   * Declaration  : void scan_tpanel(void)
184   *-------------------------------------------------------------------------------
185   * Function     : Ports PE0 and PE7 output 1 (start scanning the pen-down).
186   *              : Sets IRQ4.
187   *-------------------------------------------------------------------------------
188   * Argument     : void
189   *-------------------------------------------------------------------------------
190   * Return value : void
191   *-------------------------------------------------------------------------------
192   * Notice       : Touchscreen settling time depends on the type of the transistor
193   *              : and touchscreen. Use the actual product before deciding the
194   *              : time.
195   *""FUNC COMMENT END""*************************************************************/
196  void scan_tpanel(void)
197  {
198    /* ==== Starts scanning the pen-down ==== */
199    pen_down=0;
200
201    /* ---- Sets PE1, PE2, PE3 to OFF, PE0, PE7 to ON ---- */
202    PE.DR.BIT.B7 = 1;
203    PE.DR.BIT.B3 = 0;
204    PE.DR.BIT.B2 = 0;
205    PE.DR.BIT.B1 = 0;
206    PE.DR.BIT.B0 = 1;
207    port_wait(1000);     /* Touchscreen settling time: 1 ms */
208
209    /* ==== Sets IRQ4 ==== */
210    INTC.ICR1.BIT.IRQ4S = 1;
211    PFC.PCCRL1.BIT.PC0MD = 3;
212    PFC.PCPCRL.BIT.PC0PCR = 1;
213    INTC.IPR02.BIT._IRQ4 = 15;
214  }
215
216  #pragma section IntPRG
```

3.6 Sample Program Listing "main.c" (6/9)

```
217  /*""FUNC COMMENT""*****************************************************************
218   * ID           :
219   * Outline      : IRQ4 interrupt handling
220   *-------------------------------------------------------------------------------
221   * Include      : #include "iodefine.h"
222   *-------------------------------------------------------------------------------
223   * Declaration  : INT_IRQ4(void)
224   *-------------------------------------------------------------------------------
225   * Function     : Sets 1 to the pen_down variable.
226   *-------------------------------------------------------------------------------
227   * Argument     : void
228   *-------------------------------------------------------------------------------
229   * Return value : void
230   *-------------------------------------------------------------------------------
231   * Notice       :
232   *""FUNC COMMENT END""*************************************************************/
233  void INT_IRQ4(void)
234  {
235    pen_down=1;
236  }
237
238  #pragma section
239  /*""FUNC COMMENT""*****************************************************************
240   * ID           :
241   * Outline      : A/D conversion and coordinates calculation
242   *-------------------------------------------------------------------------------
243   * Include      : #include "iodefine.h"
244   *-------------------------------------------------------------------------------
245   * Declaration  : detect_tpanel(void)
246   *-------------------------------------------------------------------------------
247   * Function     : Ports PE0 to PE3, and PE7 apply the voltage to the touchscreen
248   *              : and the A/D Converter converts the voltage of the pen-down point.
249   *              : Calculates the coordinate of the pen-down point from the A/D
250   *              : value to store the coordinates in X_adr and Y_adr.
251   *-------------------------------------------------------------------------------
252   * Argument     : void
253   *-------------------------------------------------------------------------------
254   * Return value : void
255   *-------------------------------------------------------------------------------
256   * Notice       : Touchscreen settling time depends on the type of the transistor
257   *              : and touchscreen. Use the actual product before deciding the
258   *              : time.
259   *""FUNC COMMENT END""*************************************************************/
260  void detect_tpanel(void)
261  {
262    unsigned short ad_right;    /* ADC0 ch0 (Right) converted data */
263    unsigned short ad_bottom;       /* ADC0 ch1 (Bottom) converted data */
264    unsigned short ad_left;         /* ADC0 ch2 (Left) converted data */
265    unsigned short ad_top;          /* ADC0 ch3 (Top) converted data */
266
267    unsigned short ad_Y;        /* Y-coordinate A/D data average value */
268    unsigned short ad_X;        /* X-coordinate A/D data average value */
269
```

3.7 Sample Program Listing "main.c" (7/9)

```
270     /* ====  Controls ports to detect Y-coordinate ==== */
271     /* ---- Sets PE1, PE2, PE3 to OFF, PE0, PE7 to ON ---- */
272     PE.DR.BIT.B7 = 0;
273     PE.DR.BIT.B3 = 1;
274     PE.DR.BIT.B2 = 1;
275     PE.DR.BIT.B1 = 0;
276     PE.DR.BIT.B0 = 0;
277     port_wait(1000);     /* Touchscreen settling time: 1 ms */
278
279     /* ---- ADC0 ch0 A/D conversion ---- */
280     ADC0.ADANSR.BYTE = 0x01;        /* Specifies ADC0 ch0 */
281     ADC0.ADCR.BIT.ADST = 1;             /* Starts A/D conversion */
282     while(ADC0.ADSR.BIT.ADF == 0);{     /* Waits for A/D conversion to complete */
283     }
284     ADC0.ADSR.BIT.ADF = 0;
285     ad_right = ADC0.ADDR0;
286
287     /* ---- ADC0 ch2 A/D conversion ---- */
288     ADC0.ADANSR.BYTE = 0x04;        /* Specifies ADC0 ch2 */
289     ADC0.ADCR.BIT.ADST = 1;             /* Starts A/D conversion */
290     while(ADC0.ADSR.BIT.ADF == 0);{     /* Waits for A/D conversion to complete */
291     }
292     ADC0.ADSR.BIT.ADF = 0;
293     ad_left = ADC0.ADDR2;
294
295     /* ====  Controls ports to detect X-coordinate ==== */
296     /* ---- Sets PE2, PE3, PE7 to OFF, PE0, PE1 to ON ---- */
297     PE.DR.BIT.B7 = 0;
298     PE.DR.BIT.B3 = 0;
299     PE.DR.BIT.B2 = 0;
300     PE.DR.BIT.B1 = 1;
301     PE.DR.BIT.B0 = 1;
302     port_wait(1000);     /* Touchscreen settling time: 1 ms */
303
304     /* ---- ADC0 ch1 A/D conversion ---- */
305     ADC0.ADANSR.BYTE = 0x02;        /* Specifies ADC0 ch1 */
306     ADC0.ADCR.BIT.ADST = 1;             /* Starts A/D conversion */
307     while(ADC0.ADSR.BIT.ADF == 0);{     /* Waits for A/D conversion to complete */
308     }
309     ADC0.ADSR.BIT.ADF = 0;
310     ad_bottom = ADC0.ADDR1;
311
312     /* ---- ADC0 ch3 A/D conversion ---- */
313     ADC0.ADANSR.BYTE = 0x08;        /* Specifies ADC0 ch3 */
314     ADC0.ADCR.BIT.ADST = 1;             /* Starts A/D conversion */
315     while(ADC0.ADSR.BIT.ADF == 0);{     /* Waits for A/D conversion to complete */
316     }
317     ADC0.ADSR.BIT.ADF = 0;
318     ad_top = ADC0.ADDR3;
319
```

3.8 Sample Program Listing "main.c" (8/9)

```
320     /* ==== Starts scanning the pen-down ==== */
321     /* ---- Sets PE1, PE2, PE3 to OFF, PE0, PE7 to ON ---- */
322     PE.DR.BIT.B7  = 1;
323     PE.DR.BIT.B3  = 0;
324     PE.DR.BIT.B2  = 0;
325     PE.DR.BIT.B1  = 0;
326     PE.DR.BIT.B0  = 1;
327     port_wait(1000);    /* Touchscreen settling time: 1 ms */
328
329     if (PC.PR.BIT.B0  == 0) { /* if is to ignore the incorrect interrupt */
330                           /* when the pen-down is canceled */
331
332       Y_adr = (ADC0.ADDR0+ADC0.ADDR2)/2;/* Y-coordinate A/D data average value */
333       X_adr = (ADC0.ADDR1+ADC0.ADDR3)/2;/* X-coordinate A/D data average value */
334
335       /* ==== Sets the limit value and offsets ==== */
336       if(Y_adr > 3790){
337           Y_adr = 3790;             /* Sets the limit to the top  */
338       }
339       if(Y_adr < 225){
340           Y_adr = 225;              /* Sets the limit to the bottom  */
341       }
342       if(X_adr > 3806){
343           X_adr = 3806;             /* Sets the limit to the left  */
344       }
345       if(X_adr < 197){
346           X_adr = 197;              /* Sets the limit to the right  */
347       }
348
349       Y_adr = Y_adr-225;            /* Sets the offset to Y-coordinate A/D data */
350       X_adr = X_adr-197;            /* Sets the offset to X-coordinate A/D data */
351
352       /* ==== Converts the A/D data into the pixel coordinates ==== */
353       Y_adr = Y_adr*0.08976;
354       X_adr = X_adr*0.06650;
355
356       if(Y_adr > 319){
357           Y_adr = 319;              /* Sets the limit value  */
358       }
359       Y_adr = 319-Y_adr;            /* Y-coordinate */
360       if(X_adr > 239){
361           X_adr = 239;              /* Sets the limit value  */
362       }
363       X_adr = 239-X_adr;            /* X-coordinate */
364     }
365   }
366
```

3.9 Sample Program Listing "main.c" (9/9)

```
367  /*""FUNC COMMENT""*****************************************************************
368   * ID           :
369   * Outline      : Software wait
370   *-------------------------------------------------------------------------------
371   * Include      :
372   *-------------------------------------------------------------------------------
373   * Declaration  : void port_wait(int time);
374   *-------------------------------------------------------------------------------
375   * Description  : Wait loop
376   *-------------------------------------------------------------------------------
377   * Argument     : int time ; Wait time (time x 1 us)
378   *-------------------------------------------------------------------------------
379   * Return Value : none
380   *-------------------------------------------------------------------------------
381   * Notice       : none
382   *""FUNC COMMENT END""*************************************************************/
383  void port_wait(int time)
384  {
385    int i,j;
386    for(j=0;j<time;j++){
387      for(i=0;i<40;i++){
388          nop();
389      }
390    }
391  }
392  /* End of File */
```

4. References

- Software Manual
  SH-2A/SH2A-FPU Software Manual Rev. 3.00
  The latest version of the software manual can be downloaded from the Renesas Electronics website.

- Hardware Manual
  SH7214 Group, SH7216 Group Hardware User's Manual Rev. 2.00
  The latest version of the hardware manual can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov.29.10	—	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 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, 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 one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type 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.

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.
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

Colophon 1.0