HMI Design Using HVAC Platform

by: Chen Peng, Changhao Shi, and Alex Liu

Contents

1   Introduction

HMI refers to human-machine interface and it includes 3x3 matrix keypad, two encoder type knobs, and a segment LCD in the HVAC platform solution.

MC9S08LG32 MCU is used to implement HMI function in HVAC platform solution, and it communicates via SPI with MC9S12G240 MCU which utilizes other HVAC functions.

In addition, touch pad could replace matrix keypad and touch screen display could replace segment LCD in a typical HVAC application.

2   HMI hardware description

This section provides an introduction to the basic concepts of HMI components.

2.1   Matrix keypad

Matrix keypads are common input devices in embedded systems. They have simple architecture and are easy to interface. One of the advantages of matrix keypads is their ability to interface a large number of input keys to a microcontroller with minimum usage of I/O resources.

2.1.1 Description

Matrix keypads are simply an extension to the simple tact switch inputs. They consist of keys interconnected in the shape of a matrix. Each key is a simple mechanical switch located at the crossing between the matrix rows and columns. When a key is pressed, its row and column form an electrical contact. The rows and columns can be connected to the pins of microcontroller ports. The major advantage of using a matrix keypad is that it allows to interface a large number of keys with a relatively small number of microcontroller pins.

2.1.2 How does a key-matrix work?

For example, a 16-key keypad requires only 8 (instead of 16, if interfaced individually) I/O pins of the microcontroller if organized into a 4 rows and 4 columns matrix. The circuit diagram for this experimental tutorial is shown below.
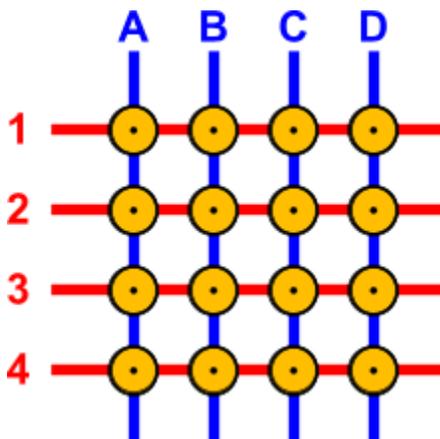

Figure 1. Matrix keypad theory

In Figure 1, the blue lines are the columns and the red lines are rows. There are 16 knots that the rows and columns intersect. It must be noted that the columns and rows are not in contact. For example, to make a key matrix, connect a button to each knot and the buttons will have a push-to-make contact.

When the operator pushes this button, it will connect the column and the row that it corresponds to. Now, put the push-to-make buttons onto the matrix.

The buttons are named with the Column:Row name that they connect. For example, the top-left button is named A1 and the bottom right is named D4.

To understand the operation principle, the user can redraw the above matrix without colors and put connection pins to each row and column wire. Then, supply power to only one column, the column B. The wire that is red, indicates that it has power, and the button that is purple indicates that the button is pressed.

The column wire B has power all the time. No other wire has power, until the button B3 is pressed. This button makes contact between the column B and the row 3. Because column B has power, the row 3 will also have power as long as the button B3 is pressed. This means that if it is known which column has currently power, and if power is detected on a row, then it is easy to understand which button is pressed. For example, if it is known that column B has power and the user also detects power on row 3, it can be assumed that the button B3 is pressed.

2.1.3 Schematic diagram

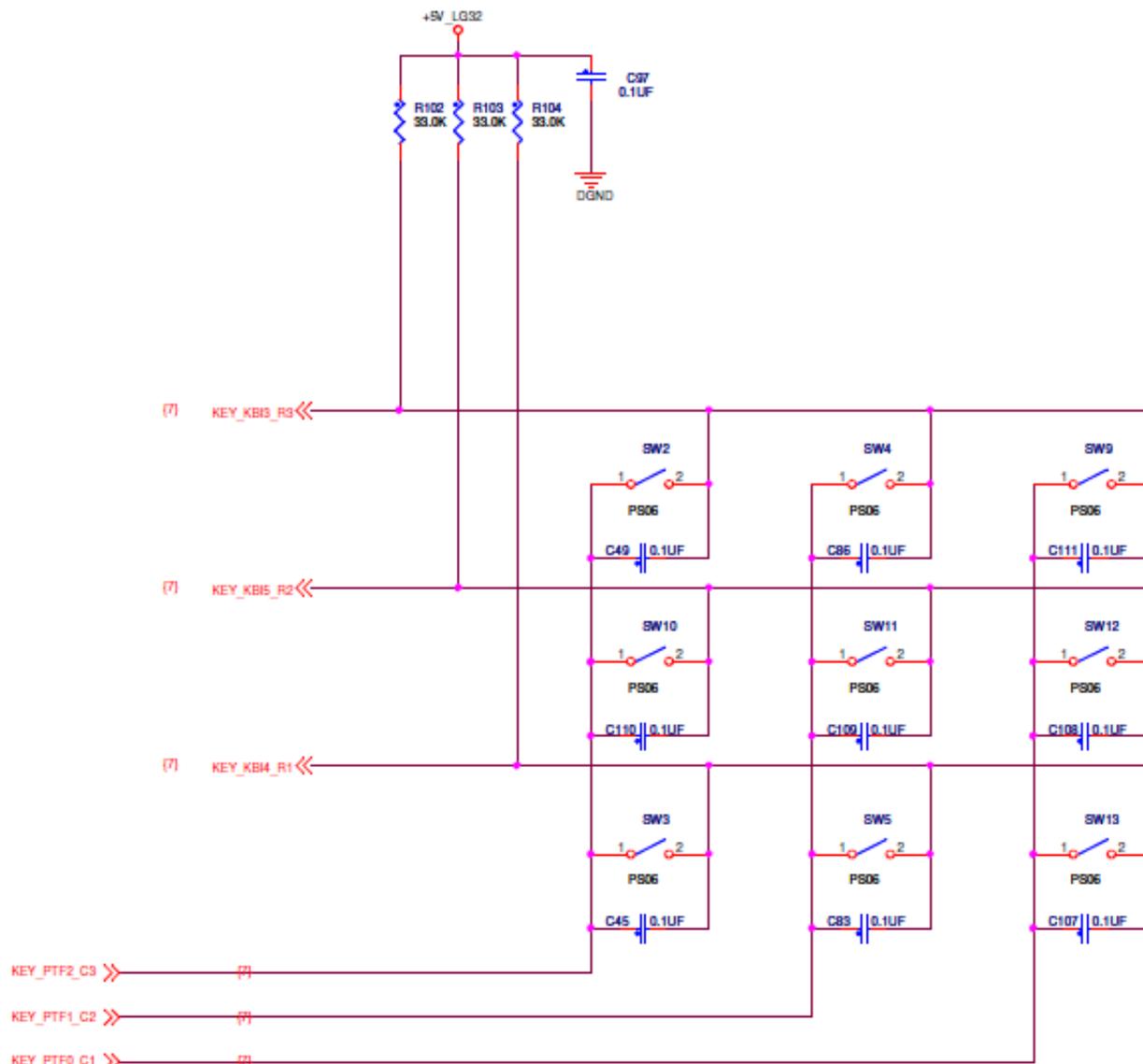The following figure presents a schematic diagram of a matrix keypad.

Figure 2. Matrix keypad schematic

2.2   Knob

2.2.1   Description

In typical HVAC systems, Knob usually could be divided into switch type and encoder type. For encoder type, it will generate two PWM output while it rotates. By detecting the PWM output phase and edge number, the MCU could get the information of rotation angle and direction.

At the same time, some encoder type knobs also have an additional key on it.

HMI Design Using HVAC Platform, Rev. 0, 10/2012

2.2.2   How does an encoder work?

The encoder output pins PIN-A and PIN-B will produce phase-different signals with different shaft rotation direction. This can be shown in the following figure.

条件 CONDITIONS		规格 SPECIFICATIONS

、B两信号输出相位差，输出波形详细见（图2/3）（虚线表示带卡点装置的上擎子处位置）

Phase-different signals   (signal A, signal B) Details shown in<fig. 2/3>

(The broken line shows detent position.)

轴回转方向 Shaft rotati- onal direction	信号 Signal	输出波形 Output
		图2 fig.2 图3 fig.3
顺时针方向 C.W	A（A-C端子间） A(Terminal A-C)	
	B（B-C端子间） B(Terminal B-C)	
逆时针方向 C.C.W	A（A-C端子间） A(Terminal A-C)	
	B（B-C端子间） B(Terminal B-C)	

Figure 3. Knob theory

2.2.3   Schematic diagram

The knob with a low-profile rotary encoder, and one push button function is used. The following figure depicts the schematic diagram of a knob.
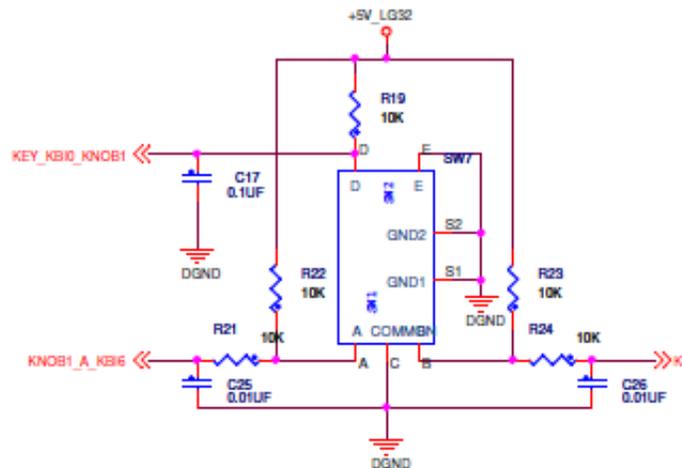
Figure 4. Knob schematic

HMI Design Using HVAC Platform, Rev. 0, 10/2012

2.3   Segment LCD

2.3.1   Description

Liquid Crystal Displays (LCD) use the liquid crystal molecules on the physical structure and optical characteristics of the display.

An LCD segment is a single bar that is used to help create a character on an LCD. For example, a clock may use a 7-segment LCD to display each of the numbers as shown in Figure 5. In the example, the number "8" is utilizing all seven segments and the number "5" is using five segments.
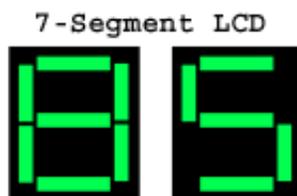

Figure 5. 7-Segments LCD

2.3.2   How does segment LCD work?

An LCD requires a pure AC drive voltage. Exposure to a DC component will reduce the life of the display significantly and must be limited to 50 mV DC. There are two types of drivers for LCDs:
- Direct drivers: A Direct drive is referred to as static and indicates that each segment of a display has an independent connection to the driver.
- Multiplex drivers: A multiplex drive (also called MUX or Duty) can simplify the connection requirements of a display with a large number of segments, for example, 16x2 dot matrix displays. Multiplexed displays always have more than one back plane.

LCDs require very less energy to operate; typically, 5–25 µA at 5 V per square inch for a twisted nematic display. However, there is a temperature coefficient, which is important for multiplexed displays. Moreover, adding a backlight or heater will necessitate additional energy.

2.3.3   Schematic diagram

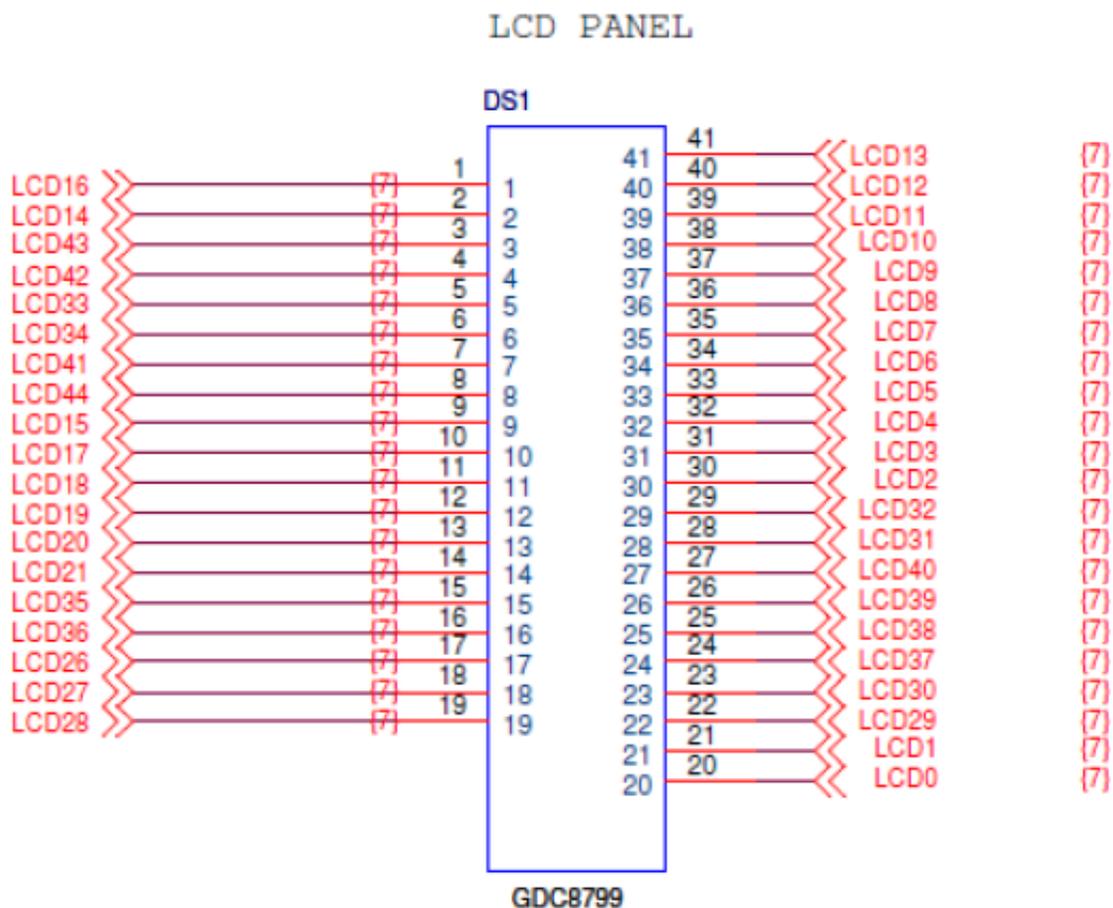The following figure shows the schematic diagram of a segment LCD.

Figure 6. Segment LCD schematic

3   HMI software description

Matrix keypad and knob are implemented using Interrupt mode. It uses 6 GPIO resources to interface 3x3 matrix keypad, and 3 GPIOs which represent rows must have the interrupt function (See Figure 1). One encoder type knob uses 2 GPIO pins and one of them must have interrupt function.

3.1   Matrix keypad

1. When a key is pressed, the Key Board Interrupt (KBI) is triggered.
2. The row is selected by interrupt source, and then set this row as output High.
3. Only one column is selected and its voltage is high. The user needs to locate one point from matrix. When the row and column of this point is found, that point could be located. This will enable KBI.

The following figure shows the Key Board Interrupt (KBI) flowchart.
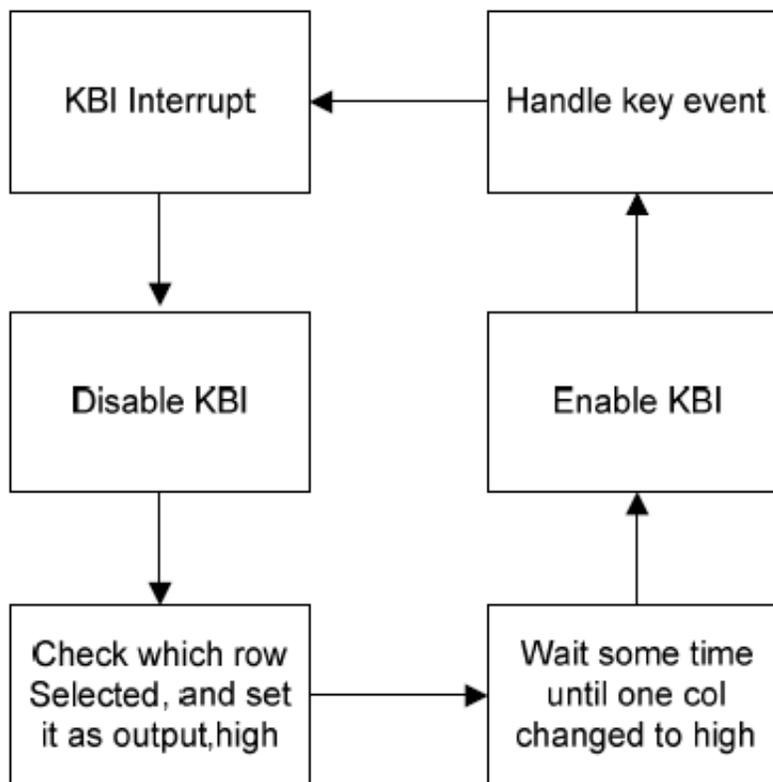
Figure 7. KBI flowchart

3.2 Knob

When user rotates the knob, it will produce the waveform and generate KBI interrupt. According to the phase difference, the rotate direction is known. According to the number of KBI edge interrupts, the overall rotation angle can be obtained.
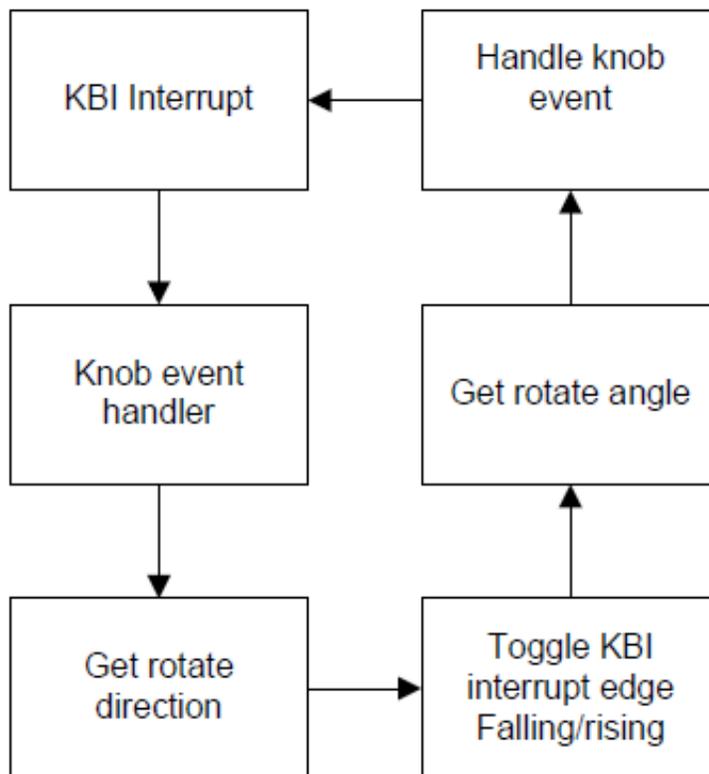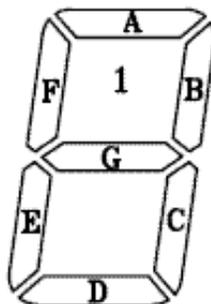
Figure 8. Knob flowchart

3.3   Segment LCD

The segment LCD (SLCD) module supports only the low-level driver. The user can decide how and when to generate displays.

3.3.1   How to show a segment

As an example, see the following figure which shows how a 7-segment LCD can be used to display a number.

The 7-segment display can be named as SF_A, SF_B, SF_C, SF_D, SF_E, SF_F, SF_G.

The following figure shows the example of an LCD PIN map, as per the LCD specs.

PIN	COM1	COM2	COM3	COM4	PIN	COM1	COM2	COM3	COM4	PIN	COM1	COM2	COM3	COM4
1	S3	7F	7G	7E	15	S18	S20	S21	S19	29	4A	4B	4C	4D
2	7A	7B	7C	7D	16	S26	S25	S24	S33	30		4F	4G	4E
3	CL3	8F	8G	8E	17	S27	S28	S29	S30	31	3A	3B	3C	3D
4	8A	8B	8C	8D	18	S23	S34	S32	S31	32	CL1	3F	3G	3E
5	S9	9F	9G	9E	19	S36	S37	S38	S39	33	2A	2B	2C	2D
6	9A	9B	9C	9D	20				COM4	34	S2	2F	2G	2E
7	S10	10F	10G	10E	21			COM3		35	1A	1B	1C	1D
8	10A	10B	10C	10D	22		COM2			36	S1	1F	1G	1E
9	S4	11F	11G	11E	23	COM1				37	COM1			
10	11A	11B	11C	11D	24	S22	S7	S6	S35	38		COM2		
11	S5	12F	12G	12E	25	6A	6B	6C	6D	39			COM3	
12	12A	12B	12C	12D	26		6F	6G	6E	40				COM4
13		L1	L2	L3	27	5A	5B	5C	5D	41	S13	S12	S8	S11
14	S17	S16	S15	S14	28	CL2	5F	5G	5E					

Figure 9. LCD pin map

In Figure 9, the COMx is backplane index and the PIN ID is LCD pin index.

For showing one segment, the user looks up the table (Figure 9), and gets the related PIN ID and COM ID. If the output is high, the corresponding segment will be shown on LCD and if the output is low, the corresponding segment will not be displayed.

3.3.2   Create pin map table

The pin map table is decided in the design phase, not in runtime phase. This means that the user needn't look up table, but get this information directly in runtime phase.

The tool 'lcdcreate' provided along with the demo code, does this job. It reads the LCD pin map with specified format, and translates it to some constant arrays.

3.3.2.1   Input for this tool

LCD pin map definition rules:
  • Input files: lcdpinmap.c and lcddrv.h.

    Output file: lcddrv.c is the low-level LCD driver code.

  • Lcdpinmap.c defines the pin map; and Lcddrv.h defines the segment name which could be used for upper-level application code.

Table 1.   LCD icon property

Icon typ	Comments
ICON_MULTI_SEG	Only used to display numbers which include multi-segment
ICON_SINGLE_SEG	Includes only one segment
ICON_FAKE	NULL segment, for example, Pin13, COM1 in Figure 9

Table 2.   LCD Pin map

Pin description	Comments
type	Please see Table 1
name	Segment name, defined in lcddrv.h
field	SF_A, SF_B, SF_C, SF_D, SF_E, SF_F, SF_G, SF_END(if this segment is not part of a number)

3.3.2.2   Creation steps

1. Define the pin map according to LCD spec as shown in the code lines below:

```
/* copy the pin map according to LCD spec */
const struct pin_map lcd_map[LCD_FIELD_NUM] =
{

    //1
    {ICON_SINGLE_SEG, ICON_S3, SF_END},
    {ICON_MULTI_SEG, NUM_POS7, SF_F},
    {ICON_MULTI_SEG, NUM_POS7, SF_G},
    {ICON_MULTI_SEG, NUM_POS7, SF_E},
    //2
    {ICON_MULTI_SEG, NUM_POS7, SF_A},
    {ICON_MULTI_SEG, NUM_POS7, SF_B},
    {ICON_MULTI_SEG, NUM_POS7, SF_C},
    {ICON_MULTI_SEG, NUM_POS7, SF_D},



enum segment_name {
    /* num must put head of enum list */
    NUM_POS1 = 0,
    NUM_POS2,
    NUM_POS3,
    NUM_POS4,
    NUM_POS5,
    NUM_POS6,
    NUM_POS7,
    NUM_POS8,
    NUM_POS9,
    NUM_POS10,
    NUM_POS11,
    NUM_POS12,
    /* icon part */
    ICON_CL1,
    ICON_CL2,
    ICON_CL3,

    ICON_FAN_L1,
    ICON_FAN_L2,
```

2. Build and run tool, it will produce the file lcddrv.c.
3. Upper application code could call APIs defined in lcddrv.c to show contents on LCD.

HMI Design Using HVAC Platform, Rev. 0, 10/2012

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com