

RL78

Software LCD driver

16-bit Single-Chip Microcontroller

RL78/G12

RL78/G13

RL78/F12

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Technology Corp. website (<http://www.renesas.com>).

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.

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 part number, confirm that the change will not lead to problems.

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

Table of Contents

Notice	2
General Precautions in the Handling of MPU/MCU Products	4
Chapter 1 Introduction	6
1.1 Overview	6
1.2 Basic function of a LCD	6
Chapter 2 Drive methods of a LCD	8
2.1 Control modes	8
2.2 Bias Voltage	9
Chapter 3 Application example	11
3.1 Triplex LCD with count down function	11
3.2 LCD timing and connection of segment and common lines.....	12
3.3 Principle schematic of this AN triplex LCD with count down function.....	13
3.4 Emulation of three level Bias Voltage.....	14
3.5 Ports of RL78/G13 (64 pin) usable for LCD drive.....	15
Chapter 4 Software Realization	16
4.1 Software description	16
4.1.1 Basic functions and flow charts	16
4.2 Program Code (LCD driving and controlling part only)	20
4.2.1 macros.h.....	20
4.2.2 lcd_timer()	21
4.2.3 main()	23
Chapter 5 Schematic and Demo Board	25
Chapter 6 Revision History	29

Chapter 1 Introduction

1.1 Overview

Today Liquid Crystal Displays (LCD) are often used in battery driven equipment which consumes as low power as possible. They are installed in a wide range of applications like CD-players, hand-held games, calculators, measuring equipment, clocks and so on.

To realise a large LCD driving as easy as possible, Renesas offers a lot of microcontrollers with on board LCD controller / driver. The user has to write the expected values into the LCD related RAM area. From then on the μC takes care of all.

Chip area for the LCD driver peripheral costs silicon and therefore money. On the other hand, depending on the application, there may be only a very small display necessary and some general purpose pins are unused. In this case it may be interesting to use these free pins to drive a small LCD.

This application note gives an example how to drive and control a LCD by usual ports, using Renesas microcontroller RL78.

1.2 Basic function of a LCD

Usually a LCD-Display is based on the "Twisted-Nematic-cell" (TN-cell): A liquid crystal mixture is included between two glass plates (substrates). These glass plates are covered with a transparent electrode-layer. The alignment of the liquid molecules is defined by direction of rubbing (the preferred direction of the substrate plates is 90 degrees to each other).

In addition both substrate plates are coated with 90° twisted polarizing filters. On the back side of this arrangement is a reflector or a transreflector. Is there a reflector like a mirror the light will be thrown back (reflexive mode). In transflexive mode a light unit has to be mounted below the glass and powered to make the displayed signs visible.

There are two modes to describe the cell into operation, the "*Normally-White-Mode*" and the "*Normally-Black-Mode*".

Normally-White-Mode (used in this application):

In this mode the cell is transparent while stand-by. If an electrical voltage is applied to the electrodes an electrical field occurs which causes a rotation of the liquid crystal molecules. These molecules are now parallel to the field and the polarizing direction of the light is no longer rotating. Therefore the second polarizing filter can not be passed and the cell gets dark.

Normally-Black-Mode:

If the polarizing filters are parallel to each other the cell will be dark when no voltage is applied and gets more transparent when voltage is applied.

All in all this cell is a voltage-controlled light-valve. A screen can be build of a lot of these cells (pixel). In this application a seven-segment display driven in normal-white-mode is used. **Figure 1, figure 2** shows one of these segments in off and on state.

Figure 1 Segment in off state

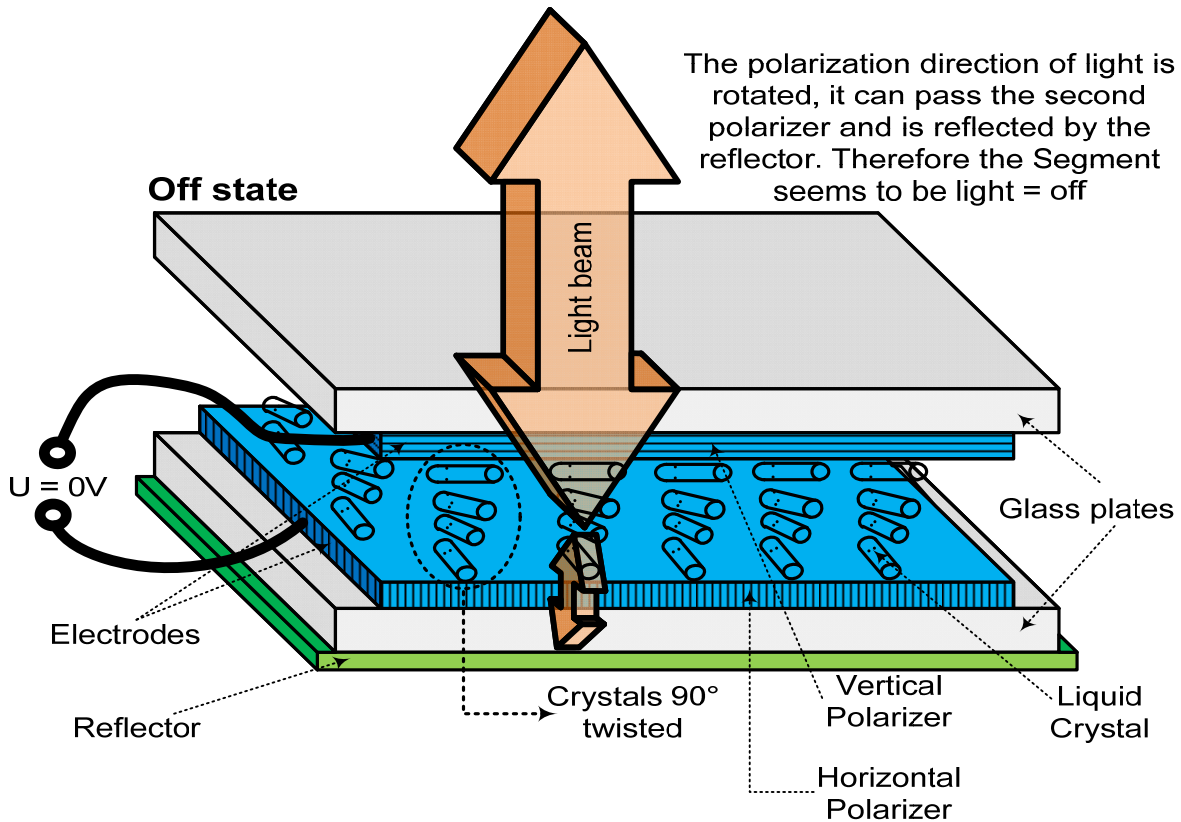
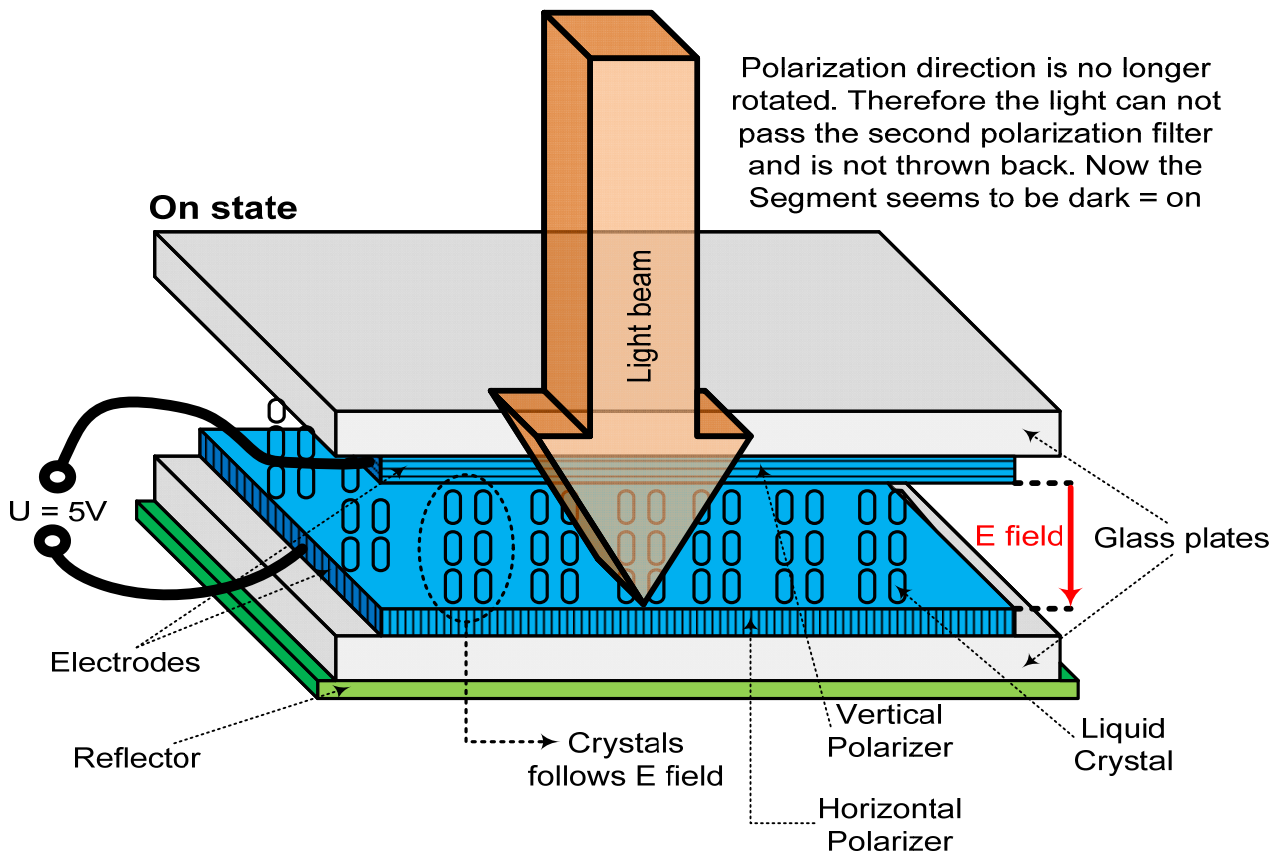


Figure 2 Segment in on state



Chapter 2 Drive methods of a LCD

2.1 Control modes

Static mode: This is the simplest mode because each segment has a separate connector (wire) and has to be driven individually. There is one backplane (common) for all segments. The disadvantage is that a lot of wires respectively ports $(n+1)$ are needed.

Duplex mode: Duplex mode is only requiring half number of wire $((n/2) + 2)$. Because of its two common lines it is called duplex mode. Every segment line can control two segments.

***Triplex mode:** The more efficient triplex mode uses three common lines and needs one third of segment lines compared to static mode $((n/3) + 3)$ only.

Quadruplex mode: Accordingly the quadruplex mode comprises four common lines with $((N/4) + 4)$ control lines in total.

The multiplex order could be extended step by step. But due to the more complex driving signal structure by different bias methods, a more multiplex level than quadruplex can hardly be achieved by port emulation.

Table 1 Control modes

Control Mode	Number of common lines	Number of segment lines	Total lines
Static mode	1	n	n + 1
Duplex mode	2	n/2	$(n/2) + 2$
*Triplex mode	3	n/3	$(n/3) + 3$
Quadruplex mode	4	n/4	$(n/4) + 4$

*This mode is used in this application note

An LCD glass withstands only a very low amount of DC level, otherwise it will be destroyed.

To avoid destruction of the liquid crystals, the voltage level at both poles of a segment has to be changed cyclic. The frequency for changing the levels depends on LCD – glasses (respectively its producers) and is usually between 30 and 300 Hz. When the change cycle is too slow, the display seems to flicker.

If it is too fast the 90° rotation of the crystals can not be completed and the segment seems to be slightly dark all time.

2.2 Bias Voltage

An individual segment on a LCD panel is visible when the potential difference of the segment and its corresponding common signal reaches a given voltage.

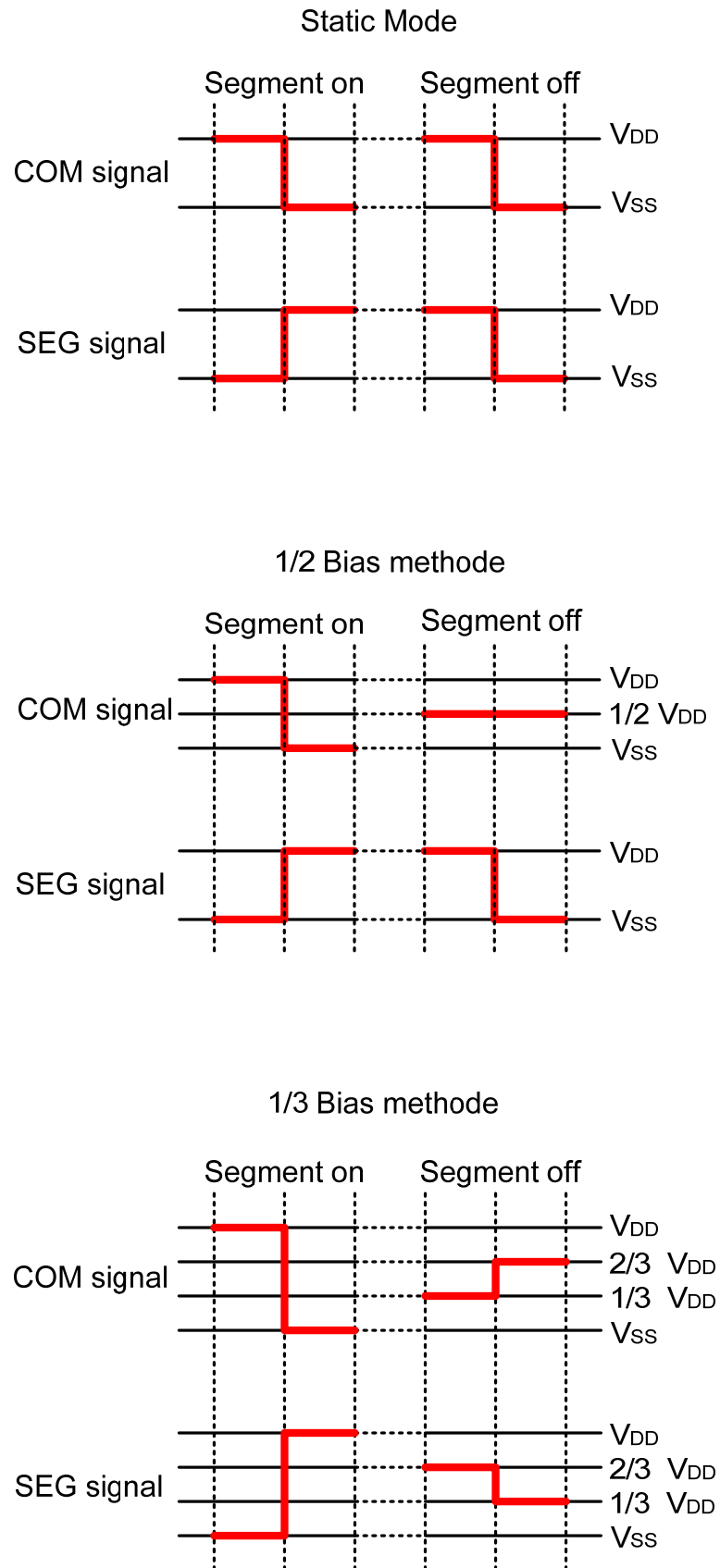
The liquid crystal will be destroyed when a DC voltage above some mV is applied to the crystals for longer time. The segment and the common voltage levels have to change cyclical, even if the LCD is static driven. However, unlike the static mode the segment lines in the multiplex modes (duplex, triplex and quadruplex) are assigned to more than one common line. Because of this the common line control must support an off-state too, where the resulting voltage between segment and common plane is low enough to keep an unselected segment switched off.

Table 2 Bias Voltages

Control Mode	BIAS Method
Static (1/1 duty)	Static
Duplex (1/2 duty)	1/2 Bias
*Triplex (1/3 duty)	1/2 Bias
	1/3 Bias
Quadruplex (1/4 duty)	1/3 Bias

*This mode is used in this application

Figure 3 Common and segment signal voltages and phases



Chapter 3 Application example

3.1 Triplex LCD with count down function

This application note describes a LCD driving example using RL78/G13 by port pins. The LCD is used to display the number 12345. When the key COUNT is pressed, there is shown a count down started with 49 down to 0 at the right two digits.

The used LCD glass is a standard TN LCD, driven in triplex mode with 36 segments. Port 1.5 – 1.7 are used for the three common lines, port 7 and port 1.0 – 1.3 do drive the segment patterns. Port 0.0 is connected to the key "COUNT". The display has to be on the whole time; therefore the μ C has to run and to drive the display cyclic all the time. This is done by internal high speed oscillation clock set to 32 MHz. This clock supplies the CPU and the peripheral clock (fCLK).

The timer channel 0 is used as an interval timer to generate a vectored interrupt each 2 ms. The interrupt routine will change LCD ports each time (for DC - suppression) and / or update the displayed values. The value of 2 ms was chosen to achieve an acceptable contrast with this specific glass. Changing the levels each 2 ms reflects in a frame frequency of about 83 Hz.

The timer channel 0 interrupt service routine is the most important part of the software, responsible for the LCD - glass driving / controlling via software. The time delay between first LCD glass wire change and last wire change has to be as fast as possible.

The watchdog timer is used for software security e.g. to detect inadvertent program loops. If such an error is detected, an internal reset signal is generated.

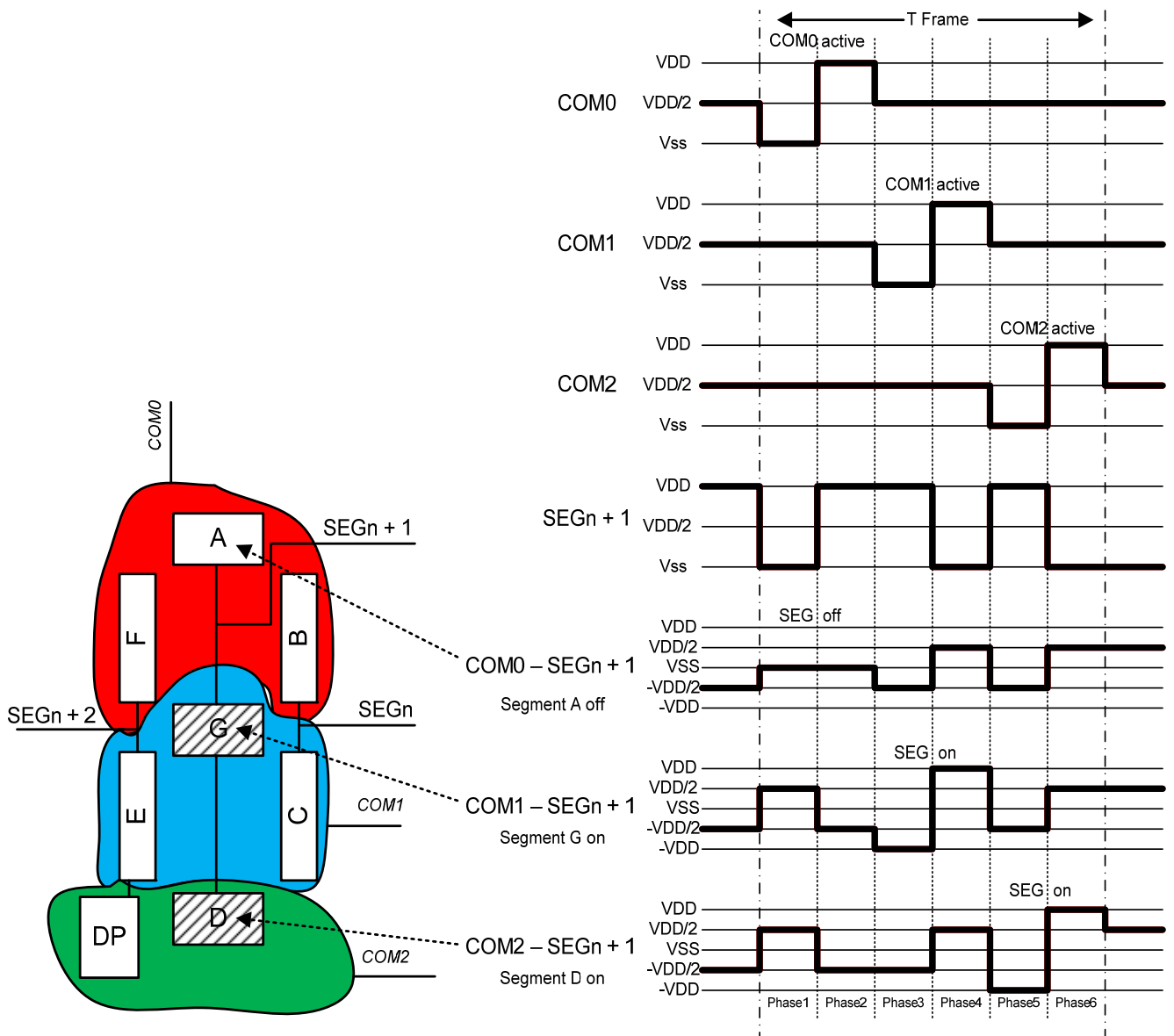
3.2 LCD timing and connection of segment and common lines

The connection and driving of a static display is very easy, because there is only one common line and only two voltage levels (VDD and GND) have to be switched to the glass. In this case, one segment line can drive one segment only.

Using triplex mode, three common lines are used, three voltage levels (VDD, 1/2 VDD and VSS) are necessary and one segment line can drive three segments. The triplex mode is more efficient; due to nearly triple number of segments can be driven compared with the static mode.

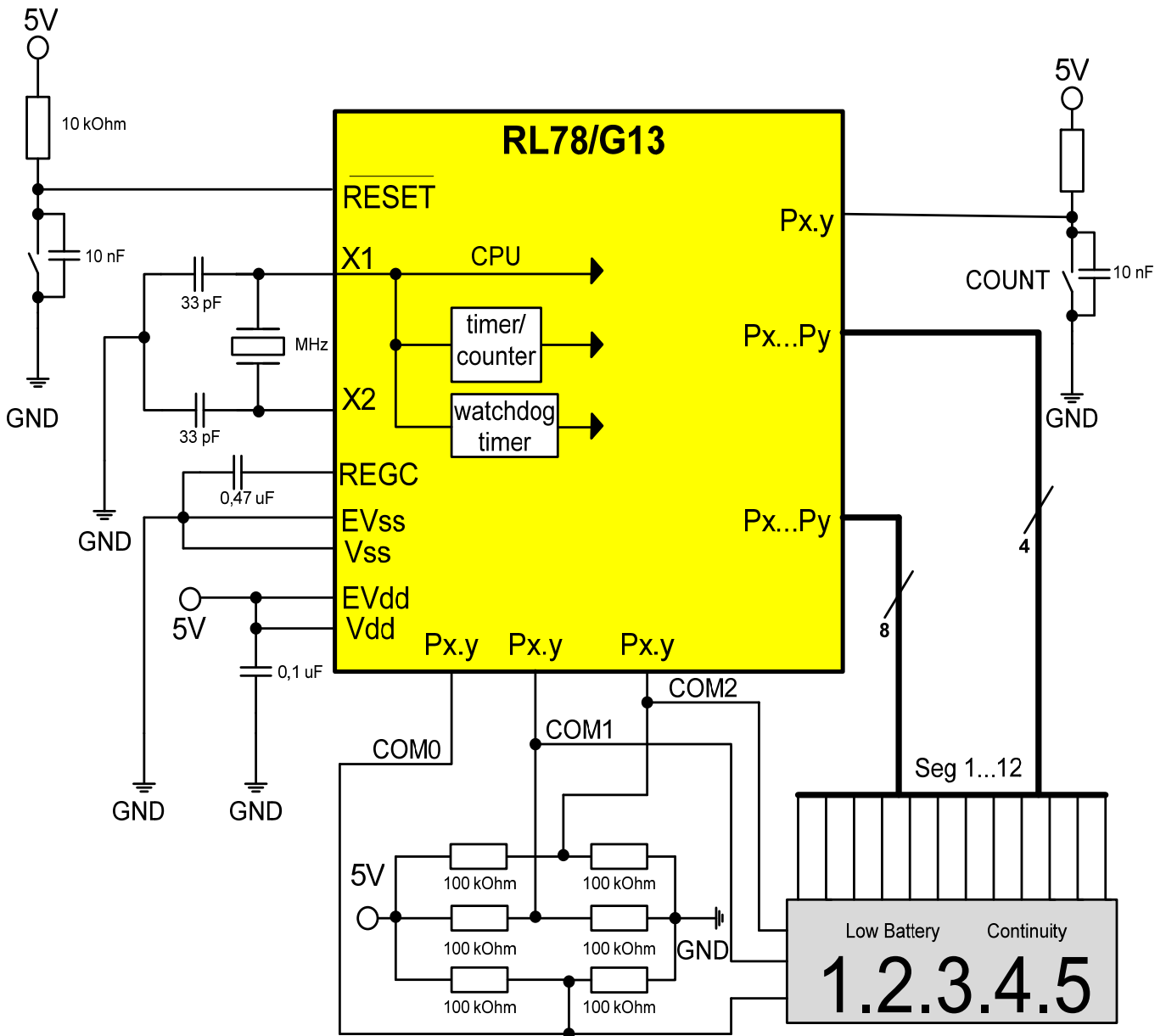
An example of LCD triplex segment connection and timing is shown in figure 4.

Figure 4 Triplex connection and timing example



3.3 Principle schematic of this AN triplex LCD with count down function

Figure 5 Principle schematic



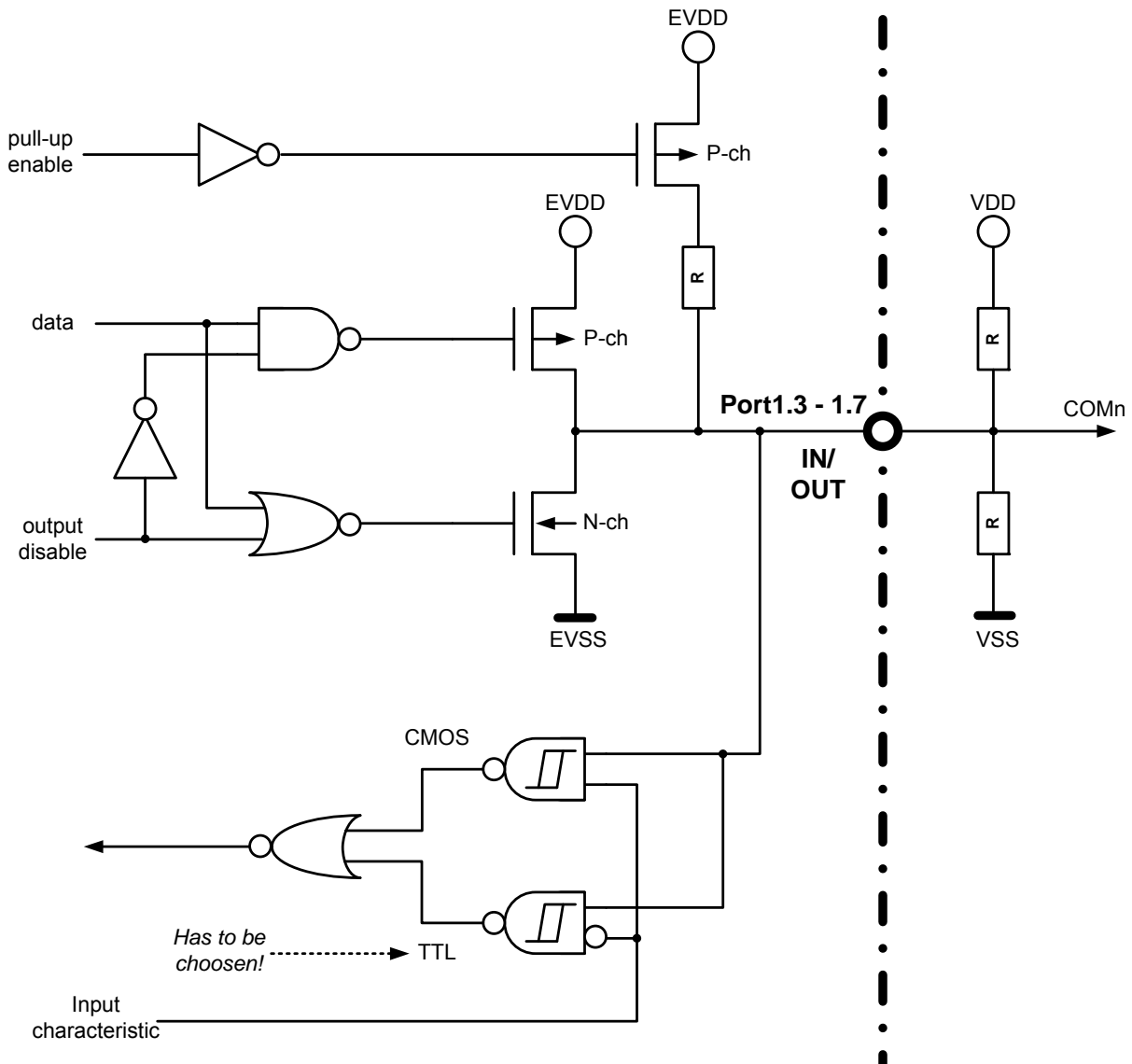
3.4 Emulation of three level Bias Voltage

General purpose output ports can not generate 1/2 VDD. Most port pins of RL78/G13 can be switched via software as input or output. In output - mode, the port can drive 5V or GND-level. When such a Port is used for common line, half VDD has to be applied to this wire additionally. This is realised by setting the port as input (high impedance) and two resistors with same resistance connected externally to that port between VDD and VSS (see figure 6).

1/2 VDD at a CMOS input is an undefined voltage level and can generate a high leakage current. To avoid this, only ports with Schmitt-Trigger TTL input should be used for common lines (RL78/G13: P01, P03, P04, P10, P11, P13-P17, P43 and P53-P55).

Furthermore, TTL level should be selected to avoid leakage current for sure.

Figure 6 Configuration of port 1.3 – 1.7 for usage as common line



3.5 Ports of RL78/G13 (64 pin) usable for LCD drive

Table 3 Which ports can be used for common- and segment line?

Port	Ports	Output high + low	After reset	Only open drain output	Pull up int. via software	IN/OUT bitwise	Schmitt Trig. Input with TTL mode	Optimal for common line	Optimal for segment line
Port 0	7	yes	input	no	yes	yes	P0.1, P0.3 - P0.4	P0.1, P0.3 - P0.4	yes
Port 1	8	yes	input	no	yes	yes	P1.0-P1.1, P1.3 - P1.7	P1.0-P1.1, P1.3-P1.7	yes
Port 2	8	yes	analog input	no	no	yes	no	no	no
Port 3	2	yes	input	no	yes	yes	no	no	yes
Port 4	4	yes	input	no	yes	yes	P43	P43	yes
Port 5	6	yes	input	no	yes	yes	P5.3 - P5.5	P5.3 - P5.5	yes
Port 6	4	only low	input	yes	no	yes	no	no	no
Port 7	8	yes	input	no	yes	yes	no	no	yes
Port 12	1	yes	input	no	yes	yes	no	no	yes
Port 13	1 only Output	yes	output	no	no	no	no	no	yes
Port 14	4	yes	input	no	yes	yes	no	no	yes

RL78/G13 (64 pin) has 41 ports useable for driving LCD segment / common lines.

Using all for LCD driving usable general purpose pins of RL78/G13, the following driving complexity can be realised:

Static Mode: 1 common and 40 segment lines.
Driving of 40 segments possible,
that means e.g. 5 Digits (7 segment) + 5 Icons.

Duplex Mode: 2 common and 39 segment lines
(1/2 Bias)
Driving of 78 segments possible,
therefore e.g. 10 Digits (7 segment) + 8 Icons can be driven.

Triplex Mode: 3 common and 38 segment lines
(1/2 Bias)
Driving of 114 segments possible,
because of that e.g. 15 digits (7 segment) + 9 icons can be displayed.

Chapter 4 Software Realization

4.1 Software description

Required resources for the example:

- Port 7, Port 1.0 - 1.3 2 ports with 12 port pins for segment lines.
- Port 1.5 – 1.7 3 port pins for common lines.
- Port 0.0 1 port pin set as input for count function.
- Timer channel 0 16-bit interval timer (~83Hz) generates interrupt.
- INTTM00 Generated timer interrupt (each 2ms).

4.1.1 Basic functions and flow charts

The most important part of the software for LCD handling is the interrupt service routine of the interval timer channel 0 which switches common and segment lines cyclic every 2 ms, called **lcd_timer()**. Task is to change between the six necessary time frames for triplex realization (each duty cycle needs two alternating states). The basic program sequence is shown in **figure 7** and the elementary interrupt service routine in **figure 8**.

For easier readability a new header file is created which defines the used ports with meaningful names, called **macros.h**.

To display the correct alphanumeric characters two 2-dimensional arrays are build having the needed values (shown in the global define of **main()**):

table7[][] for port 7 and **table1[][]** for port 1.

The corresponding time is shown in **figure 9**.

Within the main procedure of the example source code the LCD and the watchdog timer are initialized. After that initialization is completed interrupts are enabled. Furthermore the count down function (shown in **figure 7**) is defined in **main()**.

Figure 7 Basic flow chart

Initialization and operation

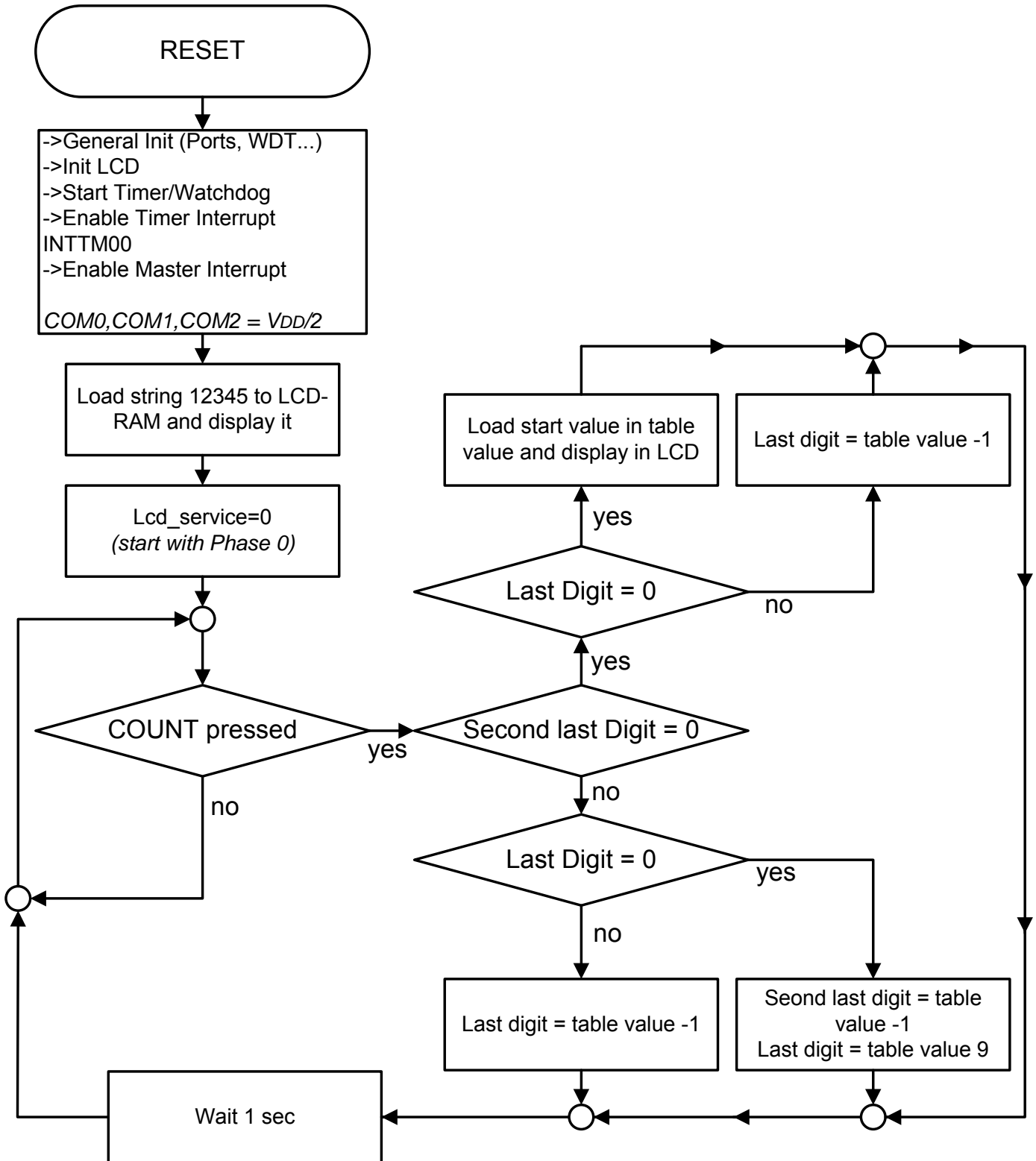


Figure 8 Principle of timer channel 0 interrupt routine

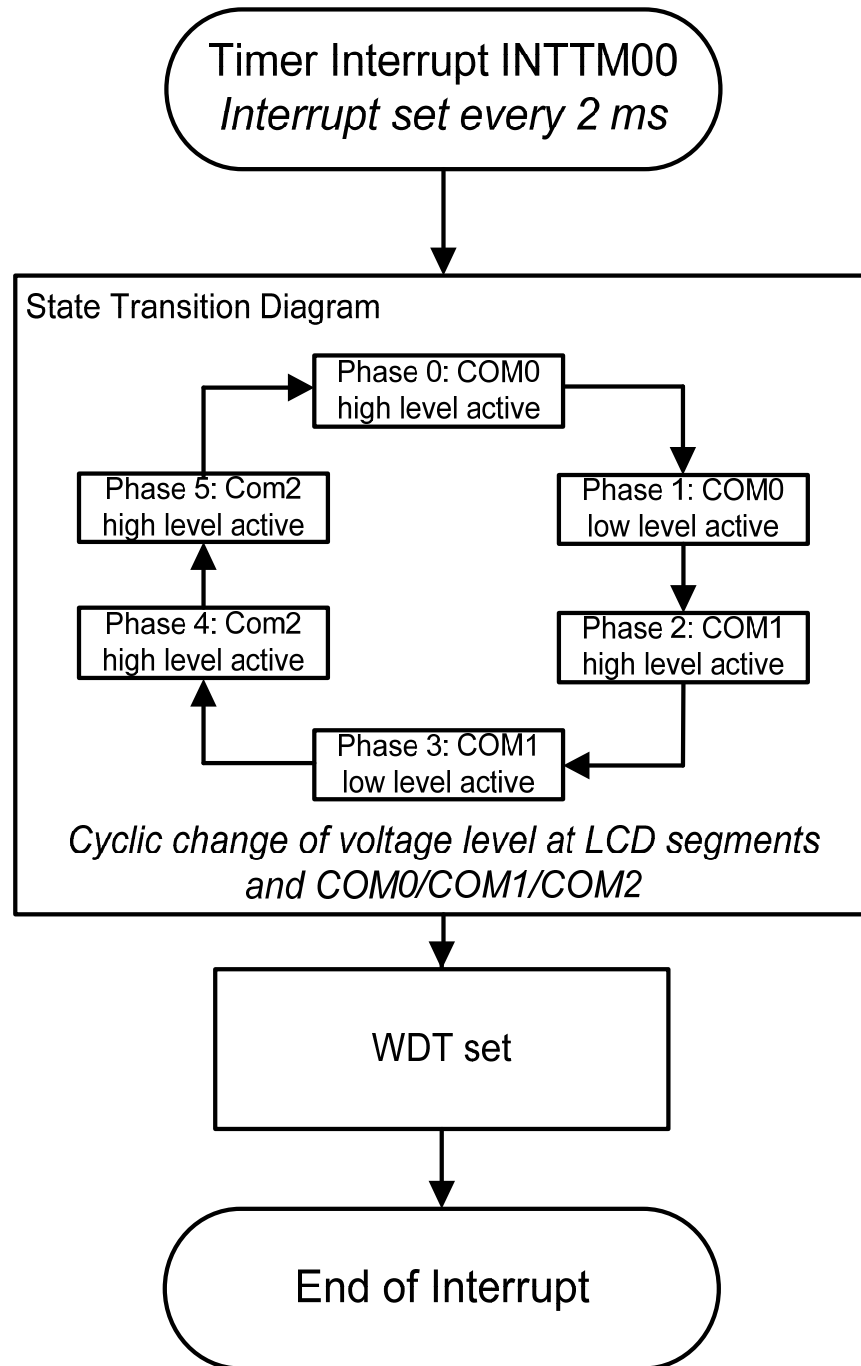
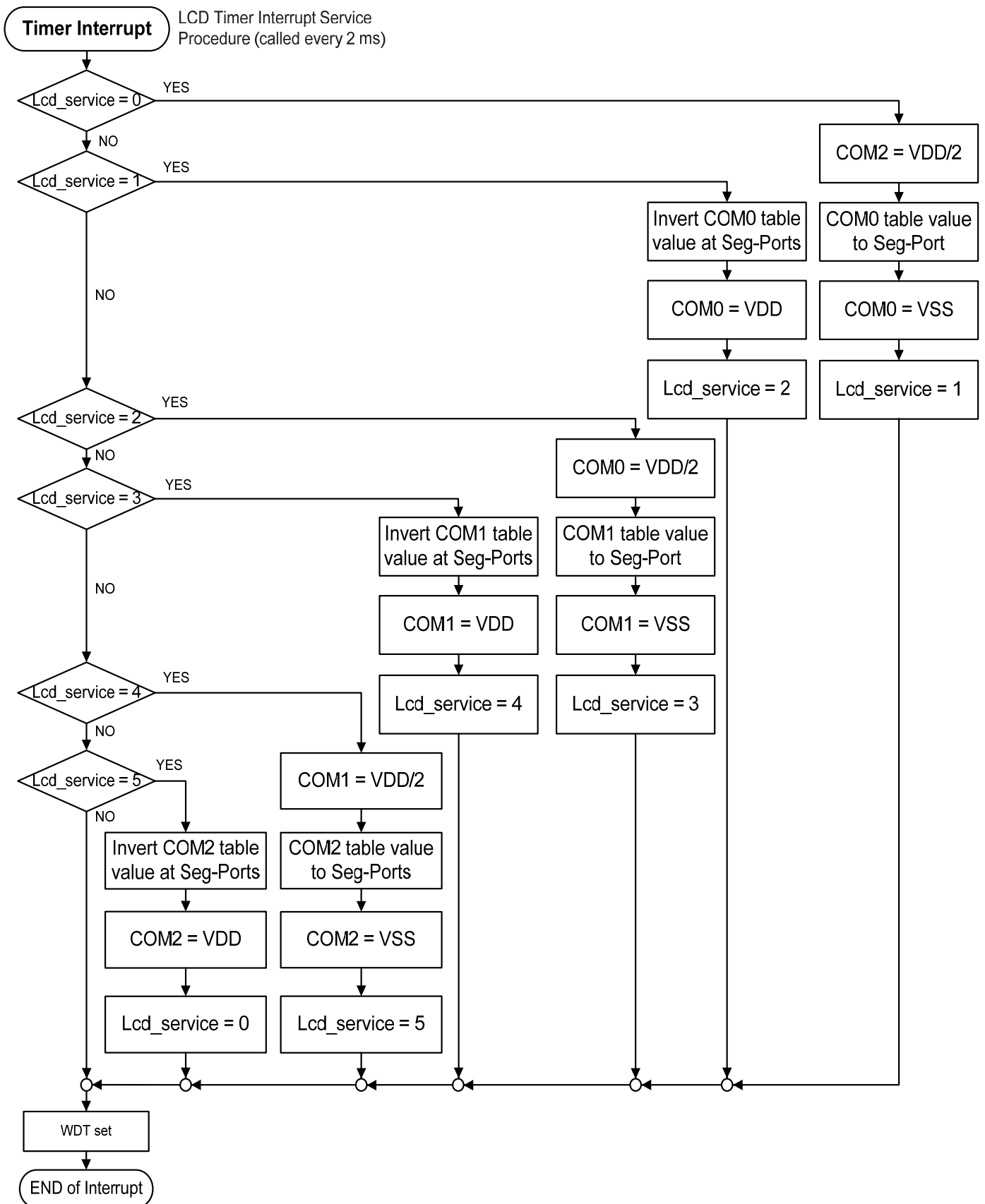


Figure 9 Flow chart of interrupt service routine



4.2 Program Code (LCD driving and controlling part only)

4.2.1 macros.h

```

#ifndef __macros_H
#define __macros_H

/*=====*/
/* Includes */
/*=====*/
/* Application Includes */

/* ----- */
/* COMs */
/* ----- */
#define com0 P1_bit.no5
#define com1 P1_bit.no6 //com'n' = P1_bit.no'm'
#define com2 P1_bit.no7
#define com0status PM1_bit.no5
#define com1status PM1_bit.no6 //com'n'status = P1_bit.no'm'
#define com2status PM1_bit.no7

/* ----- */
/* Port INPUT or OUTPUT */
/* ----- */
#define IN 1
#define OUT (!IN)

#endif /* __macros_H */

```

4.2.2 lcd_timer()

```

void lcd_timer(void)
{
  if (LCD_pause!=0) LCD_pause--;
  switch (LCD_service)
  {
    case 0: /*-----Phase_1-----*/
      com2=0; //clear COM2
      com2status=IN; //set COM2 Input
      //table value COM0 to Segment Port
      if(P0_bit.no0==1) // if P0.0 is not pressed--->display number 12345
      {
        P7=table7[10][0]; //write 12345 to Segment Port
        P1=table1[0][0];
      }
      else if (P0_bit.no0==0) //if P0.0 is pressed--->start counting
      {
        P7=( unsigned char ) COM0value; // load lower COM0 value Byte to Segment
        Port P7
        P1= * ( ( unsigned char* )&COM0value + 1 ); // load higher COM0 value Byte to
        Segment Port P1
      }
      com0=0; //clear COM0
      com0status=OUT; //set COM0 Output
      LCD_service = 1;
      break;
    case 1: /*-----Phase_2-----*/
      P7=~P7;
      P1_bit.no0=~P1_bit.no0;
      P1_bit.no1=~P1_bit.no1; //toggle Segment Ports
      P1_bit.no2=~P1_bit.no2;
      P1_bit.no3=~P1_bit.no3;
      com0=1; //Set COM0
      LCD_service = 2;
      break;

    case 2: /*-----Phase_3-----*/
      com0=0; //clear COM0
      com0status=IN; //set COM0 Input
      //table value COM1 to Segment Port
      if (P0_bit.no0==1) // if P0.0 is not pressed...display number 12345
      {
        P7=table7[10][1]; //write 12345 to Segment Port
        P1=table1[0][1];
      }
      else if(P0_bit.no0==0) //if P0.0 is pressed--->start counting
      {
        P7=( unsigned char ) COM1value; // load lower COM1 value Byte to Segment
        Port P7
        P1= * ( ( unsigned char* )&COM1value + 1 ); // load higher COM1 value Byte to
        Segment Port P1 1
      }
      com1=0; //clear COM1
      com1status=OUT; //set COM1 Output
      LCD_service = 3;
      break;
  }
}

```

```

case 3: /*-----Phase_4-----*/
    P7=~P7;
    P1_bit.no0=~P1_bit.no0;
    P1_bit.no1=~P1_bit.no1; //toggle Segment Ports
    P1_bit.no2=~P1_bit.no2;
    P1_bit.no3=~P1_bit.no3;
    com1=1; //Set COM1
    LCD_service = 4;
    break;

case 4: /*-----Phase_5-----*/
    com1=0;
    com1status=IN;
    //table value COM2 to Segment Port
    if(P0_bit.no0==1) // if P0.0 is not pressed--->display number 12345
    {
        P7=table7[10][2]; //write 12345 to Segment Port
        P1=table1[0][2];
    }
    else if(P0_bit.no0==0) //if P0.0 is pressed--->start counting
    {
        P7=( unsigned char ) COM2value; // load lower COM2 value Byte to Segment
        Port P7
        P1= * ( ( unsigned char* )&COM2value + 1 ); // load higher COM2 value Byte to
        Segment Port P1
    }
    com2=0; //clear COM2
    com2status=OUT; //Set COM2 output
    LCD_service = 5;
    break;

case 5: /*-----Phase_6-----*/
    P7=~P7;
    P1_bit.no0=~P1_bit.no0;
    P1_bit.no1=~P1_bit.no1; //toggle Segment Ports
    P1_bit.no2=~P1_bit.no2;
    P1_bit.no3=~P1_bit.no3;
    com2=1; //set COM2
    LCD_service = 0;
    break;
}WDT_Restart(); //restart watch dog timer
}

```

4.2.3 main()

```

/*
*****
** Include files
*****
*/
#include "CG_macrodriver.h"
#include "CG_system.h"
#include "CG_port.h"
#include "CG_timer.h"
#include "CG_wdt.h"
#include "macros.h"
#include "CG_userdefine.h"

/*
*****
** Global define
*****
*/
/* Set option bytes */
#pragma location = "OPTBYTE"
__root const UCHAR opbyte0 = 0xFFU;
#pragma location = "OPTBYTE"
__root const UCHAR opbyte1 = 0xFFU;
#pragma location = "OPTBYTE"
__root const UCHAR opbyte2 = 0xE8U;
#pragma location = "OPTBYTE"
__root const UCHAR opbyte3 = 0x84U;

/* Set security ID */
#pragma location = "SECUID"
__root const UCHAR secuid[10] =
{0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U, 0x00U};

//variable definition
unsigned char LCD_service = 0;
unsigned int LCD_pause;

short unsigned int COM0value;
short unsigned int COM1value; //define a 16 Bit variable for common values
short unsigned int COM2value;

//table for Segment Output
static unsigned char table7[12][3]= //output table for P7
{{0x07U, 0x05U, 0x02U}, // '0'
 {0x01U, 0x01U, 0x00U}, // '1'
 {0x03U, 0x06U, 0x02U}, // '2'
 {0x03U, 0x03U, 0x02U}, // '3'
 {0x05U, 0x03U, 0x00U}, // '4'
 {0x06U, 0x03U, 0x02U}, // '5'
 {0x06U, 0x07U, 0x02U}, // '6'
 {0x07U, 0x01U, 0x00U}, // '7'
 {0x07U, 0x07U, 0x02U}, // '8'
 {0x07U, 0x03U, 0x02U}, // '9'
 {0xEEU, 0xDBU, 0x82U}, // '345'
};
static unsigned char table1[1][3]= //output table for P1
{{0x06U, 0x0CU, 0x06U}, // '12'
};

```

```

void init_Lcd (void) //clear all used Ports and LCD RAM
{
  COM0value = 0;
  COM1value = 0; //clear complete LCD RAM (set to 0) = display off
  COM2value = 0;
  P7=0;
  P1_bit.no0=0;
  P1_bit.no1=0; //clear all Segment Ports
  P1_bit.no2=0;
  P1_bit.no3=0;
  PM7=0;
  PM1_bit.no0=0;
  PM1_bit.no1=0; //set Segment Ports to output
  PM1_bit.no2=0;
  PM1_bit.no3=0;
}
/*
**-----
**
** Abstract:
**   This function implements main function.
**
** Parameters:
**   None
**
** Returns:
**   None
**-----
*/
void main(void)
{
  DI(); //disable Interrupts
  WDT_Init(); //init watchdog timer
  init_Lcd(); //initialize LCD
  EI(); //enable Interrupts

  TAU0_Channel0_Start(); //start interval timer with timer interrupt every 2 ms

  while (1U)
  {
    if(P0_bit.no0==0) //if P0.0 is pressed start counting down (49 down to 0)
    {
      int i=9, j=4;

      for (j=4; j>=0; j--) //count down second last digit
      {
        for (i=9; i>=0; i--) //count down last digit
        {
          while (LCD_pause!=0);
          COM0value =table7[j][0]<<3 | table7[i][0];
          COM1value =table7[j][1]<<3 | table7[i][1]; //table values in Display RAM
          COM2value =table7[j][2]<<3 | table7[i][2];
          LCD_pause=500;
        }
      }
    }
  }
}

```


Chapter 5 Schematic and Demo Board

Figure 10 Base board, LCD, Battery and keys

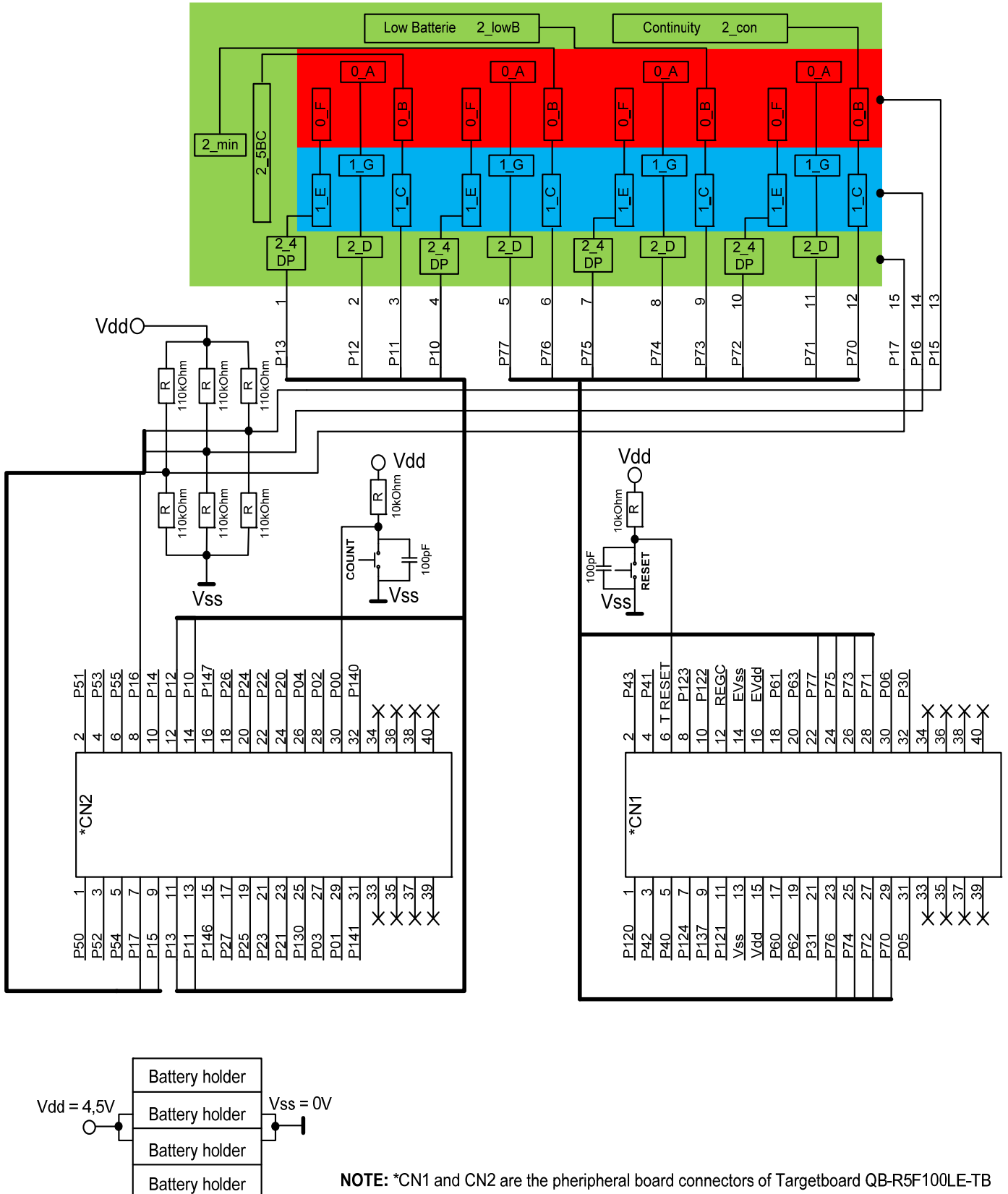


Figure 11 Microcontroller board QB – R5F100LE – TB

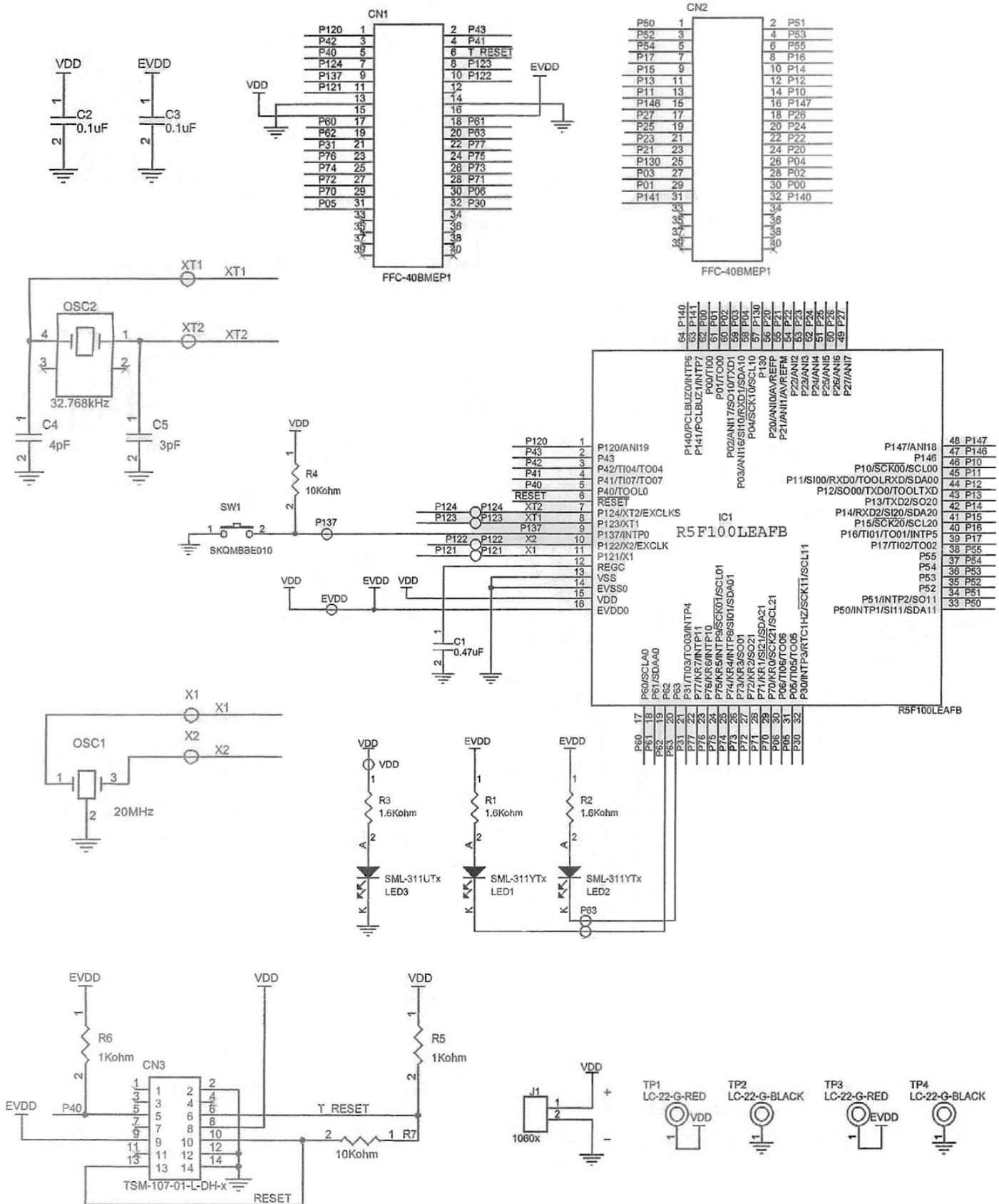
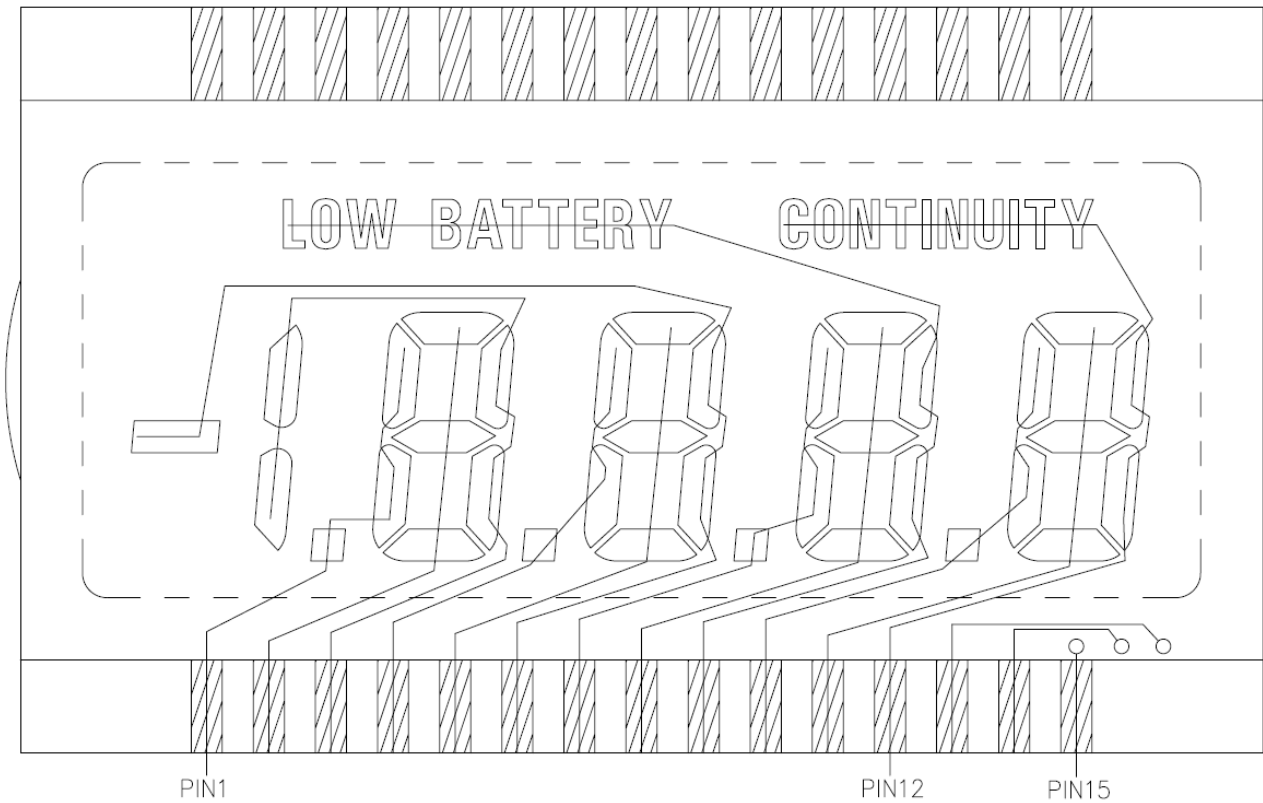
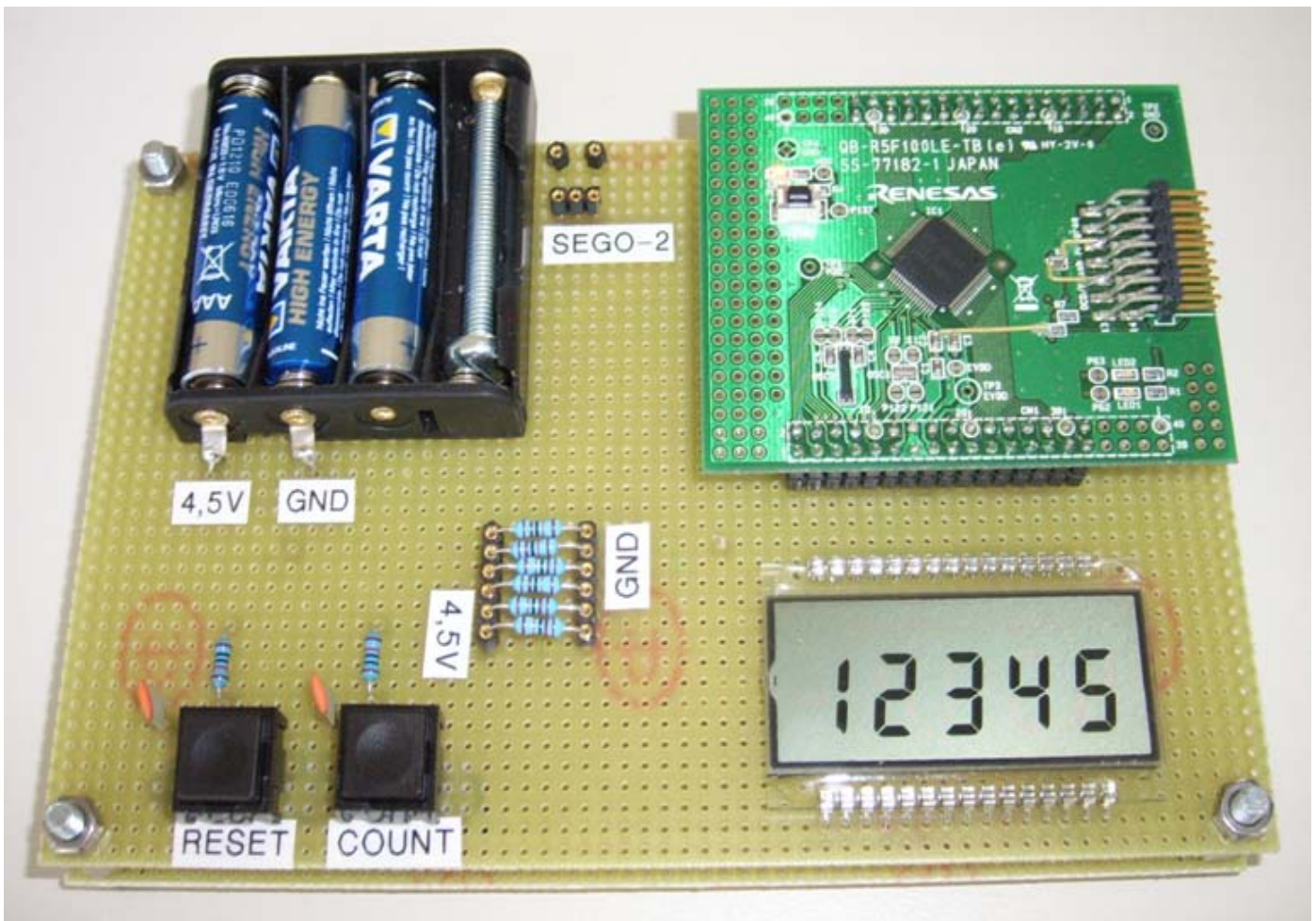


Figure 12 LCD glass and pin information



PIN	COM1	COM2	COM3
1	4F	4E	5DP
2	4A	4G	4D
3	4B	4C	5B,C
4	3F	3E	4DP
5	3A	3G	3D
6	3B	3C	Y
7	2F	2E	3DP
8	2A	2G	2D
9	2B	2C	LOW.
10	1F	1E	2DP
11	1A	1G	1D
12	1B	1C	CON.
13	COM1	----	----
14	----	COM2	----
15	----	----	COM3
16-30	NO CONNECTION		

Figure 13 Picture of the Demo Board



Chapter 6 Revision History

Table 4 Revision History

Version	Date	Change Description
0100	7. June 2011	This is the initial version of the document.



Sales Offices

Renesas Electronics Corporation

www.renesas.com

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

Renesas Electronics America Inc.

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

Renesas Electronics Canada Limited

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

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

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

Renesas Electronics Hong Kong Limited

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

Renesas Electronics Taiwan Co., Ltd.

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

Renesas Electronics Singapore Pte. Ltd.

1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.

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

Renesas Electronics Korea Co., Ltd.

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