To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

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.

Application Note

V850 Series

Dual Slope A/D Conversion

32-bit Single-Chip Microcontrollers

Document No. U17227EE1V0AN00
Date Published June 2004

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics America Inc.
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 1101
Fax: 0211-65 03 1327

Sucursal en España
Madrid, Spain
Tel: 091- 504 27 87
Fax: 091- 504 28 60

Succursale Française
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

Filiale Italiana
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

Branch The Netherlands
Eindhoven, The Netherlands
Tel: 040-244 58 45
Fax: 040-244 45 80

Branch Sweden
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

United Kingdom Branch
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
Singapore
Tel: 65-6253-8311
Fax: 65-6250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

Table of Contents

Chapter 1 Introduction 7

Chapter 2 Description of A/D Conversion Using the Dual Slope Method 8

Chapter 3 Derivation 9

Chapter 4 Circuit Diagram 10

Chapter 5 Hardware 11

5.1 Calculation of Charging Time and Timer Setup 11

5.1.1 Charging time 11

5.1.2 Timer setup 12

Chapter 6 Software 13

6.1 Main 13

6.2 Measurement 13

6.3 Data Transmission 13

Chapter 7 Software Flowcharts 14

7.1 Main Routine 14

7.2 Measurement Function 15

7.3 Data Transmission 16

Chapter 8 Listing 17

List of Figures

Figure 4-1: Circuit Diagram 10
Figure 7-1: Main Flow 14
Figure 7-2: Measurement Flow 15
Figure 7-3: Data Transmission Flow 16

Chapter 1 Introduction

This application note is written to describe an AD conversion procedure using the V850ES / Fx2 microcontroller family products. Other derivates of the V850 series can also be supported by minor changes of the source code. The AD conversion is done using the dual slope method. It is performed with a minimum of external hardware components. Only three resistors and one capacitor are needed.

Chapter 2 Description of A/D Conversion Using the Dual Slope Method

Using the dual slope method a resistor value can be converted into a digital counter value.
To do this, the charging time of a capacitor will be measured with a timer of a microcontroller. The first charging slope will use a reference resistor (R_{REF}) and the second charging slope will use variable resistor or thermistor (R_{VAR}) which should be determined. By the comparison of the two measured times and the known reference resistor (R_{REF}) the variable resistor (R_{VAR}) can be calculated.

The V850ES / Fx2 family has the advantage of bit settable I/O ports and Schmitt-trigger inputs (e.g. P32/TIP00). In this application the bit settable port P3 is used as a bidirectional port.
At first, the port P32/TI00 is cleared and set to output mode. In this case the capacitor is discharged and prepared for the first measurement. The resistor R_P is only used to limit the current during the discharging of the capacitor. Then port P33 is set to 1 and output. At this point also the 16-bit capture timer is started. The rest of the port P3 is set to input (high impedance). So the capacitor will be charged via the reference resistor R_{REF}. When the capacitor has reached the threshold level of the
Schmitt-trigger input P32/TIP00, the actual timer value is automatically captured and an internal interrupt is generated. Using this interrupt the capture value is read out. Now the capacitor will be discharged again. The same procedure starts once more with port P34. This time the capacitor is charged via the unknown resistor R_{VAR} and after the threshold is reached again the second timer value is read out.
The unknown R_{VAR} can be calculated from the two values obtained using the method described above and the value of the reference resistor. (Please refer to derivation.)

Chapter 3 Derivation

$$R_{REF}:\ V_{CREF} = V_{DD}\left(1 - e^{-\frac{t_{REF}}{R_{REF} \times C}}\right)$$

$$R_{VAR}:\ V_{CVAR} = V_{DD}\left(1 - e^{-\frac{t_{VAR}}{R_{VAR} \times C}}\right)$$

$$V_C = V_{CVAR} = V_{CREF} = const$$

The threshold level of the Schmitt-trigger input does not have any influence on the accuracy of the measurement. As this will be a constant for both measurements.

$$V_{DD}\left(1 - e^{-\frac{t_{REF}}{R_{REF} \times C}}\right) = V_{DD}\left(1 - e^{-\frac{t_{VAR}}{R_{VAR} \times C}}\right)$$

$$1 - e^{-\frac{t_{REF}}{R_{REF} \times C}} = 1 - e^{-\frac{t_{VAR}}{R_{VAR} \times C}}$$

$$e^{-\frac{t_{REF}}{R_{REF} \times C}} = e^{-\frac{t_{VAR}}{R_{VAR} \times C}}$$

$$\frac{t_{REF}}{R_{REF} \times C} = \frac{t_{VAR}}{R_{VAR} \times C}$$

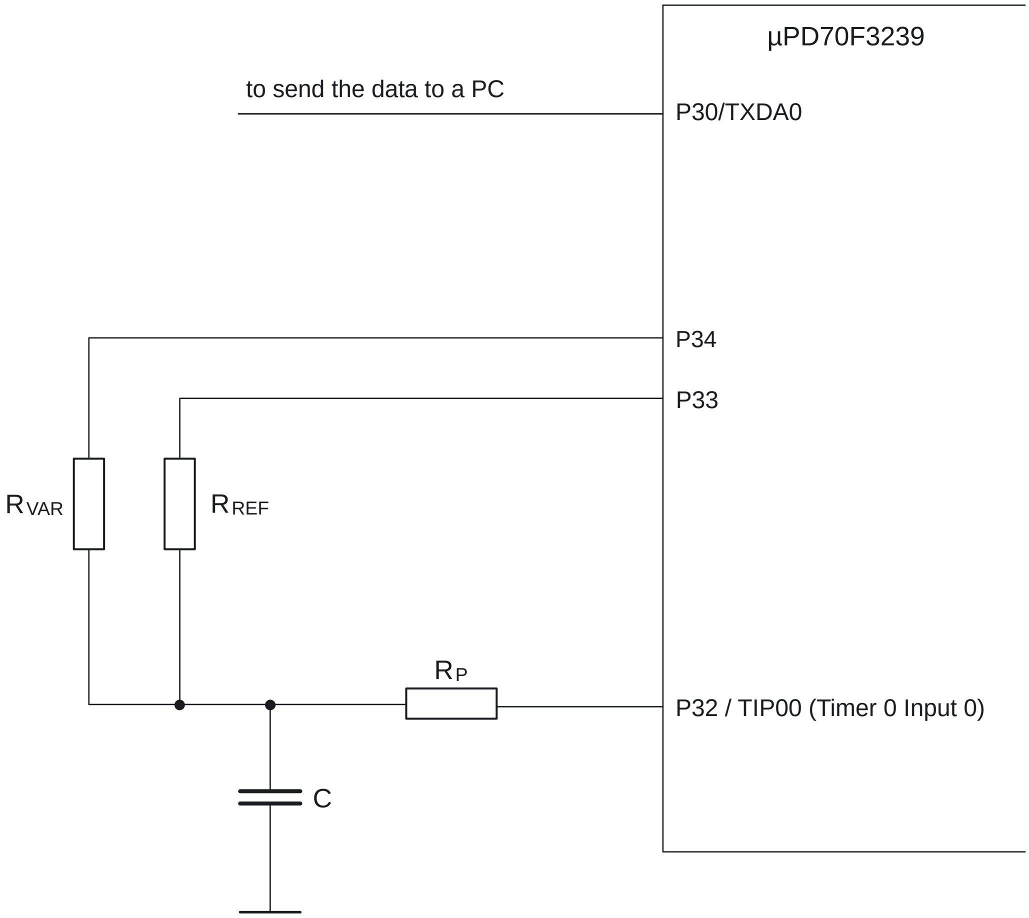
$$\frac{t_{REF}}{R_{REF}} = \frac{t_{VAR}}{R_{VAR}}$$

$$R_{VAR} = R_{REF} \times \frac{t_{REF}}{t_{VAR}}$$

C and V_{DD} do not have any influence on the accuracy of the measurement. Only the absolute value of the reference resistor R_{REF} has an influence, because these parameters will not change during one measurement. Using the R_{REF}, t_{REF} and t_{VAR}, the resistor R_{VAR} can be calculated.

Chapter 4 Circuit Diagram

Figure 4-1: Circuit Diagram



Chapter 5 Hardware

Port P33 is used to charge the capacitor via the reference resistor R_{REF}. Port P34 is used to charge the capacitor via the unknown resistor R_{VAR}. The time of charging the capacitor is measured in both cases.
Port 32 / TIP00 is used as a capture input for the timer. This port has a Schmitt-trigger behaviour. Additionally this port is used to discharge the capacitor. The protection resistor R_P will limit the discharging current and must be considered for the discharge time.
Port 30 is used as output. This pin is shared with the transmit pin of the UARTA, and used to send the measured data to a PC (terminal program).

5.1 Calculation of Charging Time and Timer Setup

5.1.1 Charging time

$$V_C = V_{DD}\left(1 - e^{-\frac{t}{R \times C}}\right)$$

$$\frac{V_C}{V_{DD}} = 1 - e^{-\frac{t}{R \times C}}$$

$$1 - \frac{V_C}{V_{DD}} = e^{-\frac{t}{R \times C}}$$

$$-\frac{t}{R \times C} = \ln\left(1 - \frac{V_C}{V_{DD}}\right)$$

$$t = -R \times C \times \ln\left(1 - \frac{V_C}{V_{DD}}\right)$$

Example:

V_{DD} = 5 V; $V_{threshold} = V_C = (0.4 \ldots 0.7)\ V_{DD}$

Typical $V_{threshold} = 0.6\ V_{DD}$

R_{REF} = 51 KΩ; C = 220 nF

R_P = 100 Ω

$t = -R_{REF} \times C \times \ln(1 - V_C/V_{DD})$

$t_{TYP} = -51\ K\Omega \times 220\ nF \times \ln(1 - 0.6)$

t_{TYP} = 10.28 ms

5.1.2 Timer setup

Max setup time:

t_{MAX} = -51 KΩ × 220 nF × ln (1 - 0.7)
t_{MAX} = 13.5 ms

Resolution:

Resolution = t_{MAX} / step time

With the 16-bit capture timer of the V850ES / Fx2 family a resolution of 50 ns (f_{XX} @20 MHz) can be realized. Thus the reached resolution will be:

Resolution = t_{MAX} / 50 ns
= 13.5 ms / 50 ns
= 270.000 steps

This amount of steps would generate an overflow of the 16-bit timer (65536 steps). So, the timer prescaler is set to a resolution of 200 ns (f_{XX}/4 @20 MHz).Then the reached resolution will be:

Resolution = t_{MAX} / 200 ns
= 13.5 ms / 200 ns
= 67.500 steps

Thus in this application note a resolution of about 16 bit is used. Due to the max timer resolution of 16-bit (65536 steps) a higher or a lower resolution can be realized by changing the timer prescaler or the setup time.

Note: It is recommended that the value of the resistors R_{REF} and R_{VAR} should be in a range between 10 KΩ and 100 KΩ.
Lower resistors than 10 KΩ may result a lower accuracy due to the voltage drop (V_{OH} = f (I_{OH})).
Higher resistor values than 100 KΩ might cause a lower accuracy due to some leakage current.

Chapter 6 Software

The software consists of the following three modules.

6.1 Main

This module contains the system initialization and the main loop.
CPU clock, all ports, the used 16-bit timer and the UART will be initialized. In the main loop the conversion is started, the result converted and the transmission module was called.

6.2 Measurement

This module will do a complete dual slope measurement.
This is in detail:

- Discharging the capacitor.
- Reference measurement using R_{REF}.
- Store the result for the reference resistor.
- Discharging the capacitor.
- Absolute measurement using R_{VAR}.
- Store the result for the variable resistor.

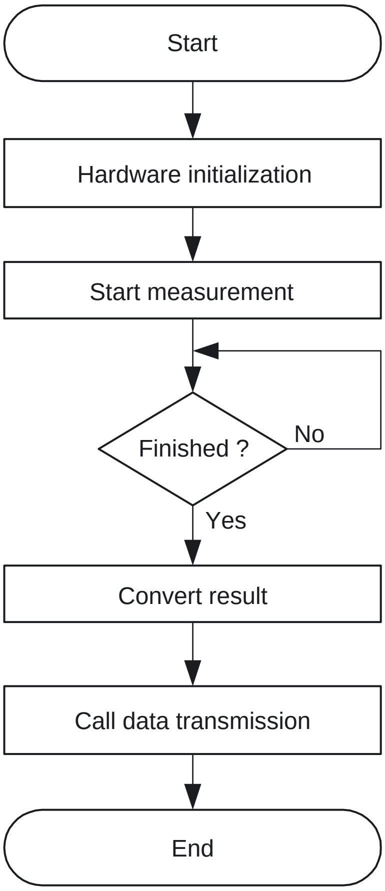
6.3 Data Transmission

This module converts the measured data to an ASCII format and sends it via UART. Using this module the measured data can be displayed on a PC using a terminal program.

Chapter 7 Software Flowcharts

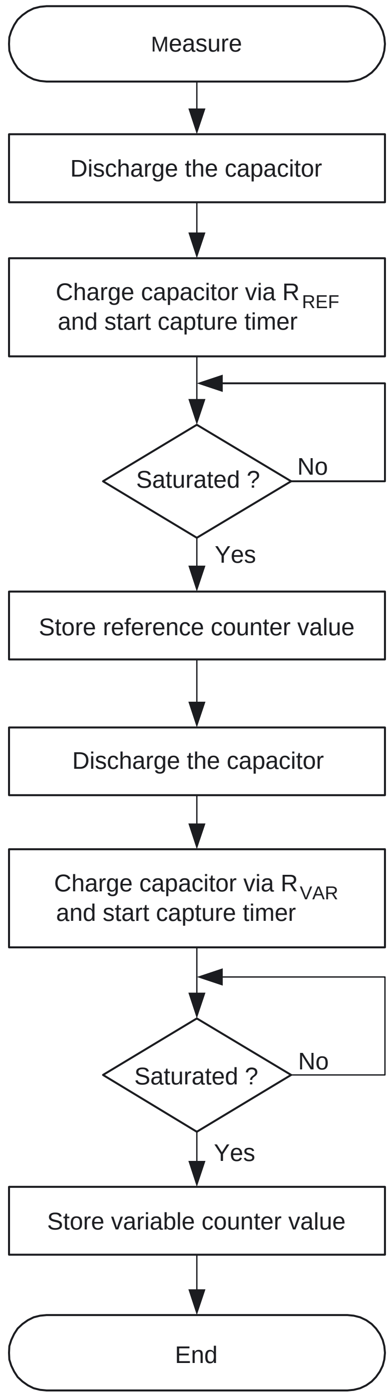
7.1 Main Routine

Figure 7-1: Main Flow



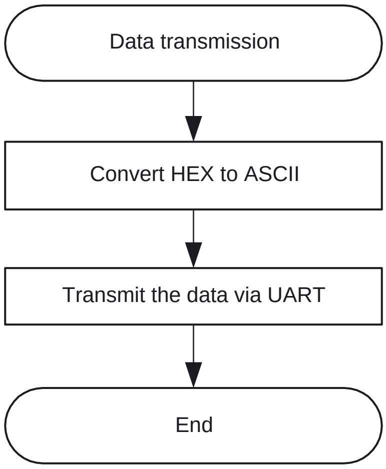
7.2 Measurement Function

Figure 7-2: Measurement Flow



7.3 Data Transmission

Figure 7-3: Data Transmission Flow



Chapter 8 Listing

```
/*==============================================================
** PROJECT      = Dual_Slope.c
** MODULE       =
** VERSION      = 0.0
** DATE         = 3.05.2004
** LAST CHANGE  =
**==============================================================
** Description: Demoprorgam for dual slope emulation
**
**==============================================================
** Enviroment:  Device:         uPD70F3239
**              C-Compiler:     Green Hills    Version 3.5.1
**
**==============================================================
** By:               NEC Electronics (Europe) GmbH
**                   Arcadia Strasse 10
**                   D-40472 Duesseldorf
**
**                   NEC-EE, CES
**==============================================================
Changes:
**==============================================================
*/

/*==============================================================
** include
**==============================================================
*/
#include "df3239.h"
#include "df3239extIO.h"

/*==============================================================
** type definitions
**==============================================================
*/
typedef enum { false = 0, true = 1 } bit;


/*==============================================================
** definitions
**==============================================================
*/
/* Reference resistor value */
#define REF_RESISTOR            56000

/* Loop constant for discharging C */
#define WAITLOOP                2000


/* macros for portpin switching */
#define TM_PORT                 ((volatile struct bitf *)(&P3))->bit02
#define TM_PORT_MODE            ((volatile struct bitf *)(&PM3))->bit02
#define TM_PORT_MODE_CONTROL    ((volatile struct bitf *)(&PMC3))->bit02
#define REF_PORT                ((volatile struct bitf *)(&P3))->bit03
#define REF_PORT_MODE           ((volatile struct bitf *)(&PM3))->bit03
```

```
#define VAR_PORT                ((volatile struct bitf *)(&P3))->bit04
#define VAR_PORT_MODE           ((volatile struct bitf *)(&PM3))->bit04

/* macros for starting and stopping the timer */
#define Timer_Start()      (((volatile struct bitf *)&(TP0CTL0))->bit07 = 1)
#define Timer_Stop()       (((volatile struct bitf *)&(TP0CTL0))->bit07 = 0)


/* State machine defines for timer ISR */
#define DISCHARGE_STATE         0
#define REF_CONV_STATE          1
#define MEASUREMNT_STATE        2


/*===============================================================
** variable definitions
**===============================================================
*/
unsigned int    ref_time, meas_time, calc_value;
unsigned char   DS_status;
bit             DS_Ready, DS_OVF_error, sendokflag;


/* ------------------------------------------------------------------ */
/* Hardware initialization                                              */
/* ------------------------------------------------------------------ */
void hdwinit (void) {

    /* Basic configuration for µPD70F323x */
    VSWC    = 0x01;     /* Peripheral wait states as recommended in manual */

    PRCMD  = 0x00;     /* Prepare to write specific register */
    PCC    = 0x00;     /* Select cpu clock fxx = foscl       */

    WDTM2   = 0x1f;     /* Stop watchdog timer                            */
    RCM     = 0x01;     /* Stop ring oscillator                           */
    PLLCTL  = 0x03;     /* F-Line CPUs have a PLL -> fxx = Osc_Freq * 4  */

    /* Port setting; Set all unused port to output to avoid floating */
    PM0    = 0x00;
    PM1    = 0x00;
    PM3H   = 0x00;
    PM4    = 0x00;
    PM5    = 0x00;
    PM6    = 0x0000;
    PM7L   = 0x00;
    PM7H   = 0x00;
    PM8    = 0x00;
    PM9    = 0x0000;
    PM12   = 0x00;
    PMDL   = 0x0000;
    PMCS   = 0x00;
    PMCT   = 0x00;
    PMCM   = 0x00;
    PMCD   = 0x00;
```

```
    /* Configure ports for dualslope */
    P3L    = 0x0C;   /* REF_PORT, VAR_PORT =1;  TM_PORT=0 */
    PM3L   = 0xFA;   /* REF_PORT, VAR_PORT =input;  TM_PORT=output */
    PMC3L  = 0x01;   /* TXDA0 output; TM_PORT=I/O(not peripheral) */
    PFC3L  = 0x00;   /* P32 = TIP00 */
    PFCE3L = 0x04;   /*      "      */

    /* 16-bit timer setting */
    //TP0CTL0  = 0x01;  /* fxx/2   */
    //TP0CTL0  = 0x02;  /* fxx/4   */
    TP0CTL0  = 0x03;    /* fxx/8   */
    //TP0CTL0  = 0x04;  /* fxx/16  */
    //TP0CTL0  = 0x05;  /* fxx/32  */
    //TP0CTL0  = 0x06;  /* fxx/64  */
    //TP0CTL0  = 0x07;  /* fxx/128 */
    TP0CTL1  = 0x05;    /* Free running mode */
    TP0IOC0  = 0x00;    /* No output */
    TP0IOC1  = 0x01;    /* Rising edge input */
    TP0IOC2  = 0x00;    /* No edge detection */
    TP0OPT0  = 0x10;    /* TP0CCR00 as capture register */
    TP0CCIC0 = 0x00; /* Enable Timer P0 capture int. 0; Priority 0 */
    TP0OVIC  = 0x00; /* Enable Timer P0 overflow err. int.; Priority 0 */

    /*UARTA0 setting */
    UA0CTL0 = 0xD2; /* Enable,Trans.,LSB first,8bit,no parity,1stop */
    UA0CTL1 = 0x02; /* 38400 bps @ 20 MHz  Error: 0.16% */
    UA0CTL2 = 65;   /*                 "               */
    UA0TIC  = 0x07; /* Enable transmit interrupt; priority 7 (lowest) */
}


/* ---------------------------------------------------------------------- */
/* Discharging capacitor for starting measurement                         */
/* ---------------------------------------------------------------------- */
void discharge_C (void) {
    unsigned int i;

    REF_PORT_MODE        = 1;  /* Switch reference port to Hi-z(Input)*/
    VAR_PORT_MODE        = 1;  /* Switch variable port to Hi-z(Input) */

    TM_PORT_MODE_CONTROL = 0; /* Switch port pin to I/O              */
    TM_PORT              = 0; /* Switch port pin to 0 -> Discharge C */

    for (i=0;i<WAITLOOP;i++) { /* Wait for discharging the capacitor  */
        __asm("nop");          /*    Waste time ...                   */
    }

    TM_PORT_MODE_CONTROL = 1;  /* Switch port pin to timer input      */
 }


/* ---------------------------------------------------------------------- */
/* Send data via UARTA0 as decimal value                                  */
/* ---------------------------------------------------------------------- */
void senddata(unsigned int data) {
    sendokflag = false;
    UA0TX = ':';
    while (sendokflag == false);
```

```
    sendokflag = false;
    UA0TX = ((data/100000)%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = ((data/10000)%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = ((data/1000)%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = '.';    /* decimal point -> xxx.xxx */
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = ((data/100)%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = ((data/10)%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = (data%10)+'0';
    while (sendokflag == false);
    sendokflag = false;
    UA0TX = 0x0D;
    while (sendokflag == false);
}


/* ---------------------------------------------------------------------- */
/* Send data via UARTA0 as decimal value                                   */
/* ---------------------------------------------------------------------- */
void sendstring(const char *str) {
    while (*str) {
        sendokflag = false;
        UA0TX = *str++;
        while (sendokflag == false);
    }
}


/* ---------------------------------------------------------------------- */
/* UARTA0 transmission end interrupt service routine                       */
/* ---------------------------------------------------------------------- */
__interrupt void INTUA0T(void) {
    sendokflag = true;
}


/* ---------------------------------------------------------------------- */
/* Timer P0 overflow interrupt service routine                             */
/*   State machine for dualslope measurement                               */
/* ---------------------------------------------------------------------- */
__interrupt void INTTP0OV(void) {
    Timer_Stop();           /* Stop and clear timer         */
    DS_OVF_error = true;    /* Set overflow error flag      */
    DS_Ready     = true;    /* Indicate end of conversion   */
}
```

```
/* ---------------------------------------------------------------------- */
/* Timer P0 capture 0 interrupt service routine                            */
/*   State machine for dualslope measurement                               */
/* ---------------------------------------------------------------------- */
__interrupt void INTTP0CC0(void) {
    unsigned int CR_value;

    CR_value = TP0CCR0;     /* Read captured timer value */
    Timer_Stop();           /* Stop and clear timer      */

    switch(DS_status) {

        case DISCHARGE_STATE:     /* Start of reference measurement */
            discharge_C();        /*   Discharging C */
            REF_PORT      = 1;    /*   Set ref. resistor port to high */
            REF_PORT_MODE = 0;    /*                  "               */
            Timer_Start();        /*   Start timer */
            DS_status = REF_CONV_STATE;
            break;

        case REF_CONV_STATE:      /* Start of 2. conversion */
            ref_time = CR_value;
            discharge_C();        /*   Discharging C */
            VAR_PORT      = 1;    /*   Set var. resistor port to high */
            VAR_PORT_MODE = 0;    /*                   "              */
            Timer_Start();        /*   Start timer */
            DS_status = MEASUREMNT_STATE;
            break;

        case MEASUREMNT_STATE:    /* End of dualslope conversion  */
            meas_time     = CR_value;
            VAR_PORT_MODE = 1;    /*   Var. res. port to Hi-z(Input)*/
            DS_Ready      = true; /*   Conversion finished */
            break;
        }
}




/* ====================================================================== */
/* main function                                                           */
/* ====================================================================== */

void main(void) {

    //DI();         /* Interrupt disable   */
    hdwinit ();     /* Peripheral settings */
    EI();           /* Interrupt enable    */

    /* Initialize dual-slope flags and status-counter */
    DS_status    = DISCHARGE_STATE;
    DS_Ready     = false;
    DS_OVF_error = false;

    /* Start conversion by enabling the timer interrupt     */
    /* and setting the interrupt flag; Priority 0 (highest) */
```

```
    TP0CCIC0 = 0x80;

    while(1) {      /* Endless loop - Main loop */

        if (DS_Ready) { /* Conversion finished */

            if (DS_OVF_error == false) {
                /* Calculate value here reference resistor = 51k */
                calc_value = REF_RESISTOR * meas_time / ref_time;
                /* Send data via UART */
                senddata(calc_value);

            } else {    /* Timer overflow error */
                if (DS_status == REF_CONV_STATE)
                    sendstring("Reference error\r");
                else /* DS_status == MEASUREMNT_STATE */
                    sendstring("Measurement error\r");
            }

            /* Initialize dual-slope flags and status-counter */
            DS_status    = DISCHARGE_STATE;
            DS_Ready     = false;
            DS_OVF_error = false;

            /* Start conversion */
            TP0CCIC0 = 0x80;
        }


        /* . . .                                                        */
        /* Do anything while conversion is in background progress ... */
        /* . . .                                                        */

    }
}
```

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics America Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49(0)-211-6503-1344

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-6250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: ______________________

Document number: ______________________ Page number: ____________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	☐	☐	☐	☐
Technical Accuracy	☐	☐	☐	☐
Organization	☐	☐	☐	☐

CS 99.1

[MEMO]