Software De-bounce

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	When I/O detection, use SysTick to count. If there is a bounce between a certain period of time, do not respond to avoid the noise.
BSP Version	NUC123 Series BSP CMSIS v3.01.001
Hardware	NuTiny-EVB-123-LQFP64_V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

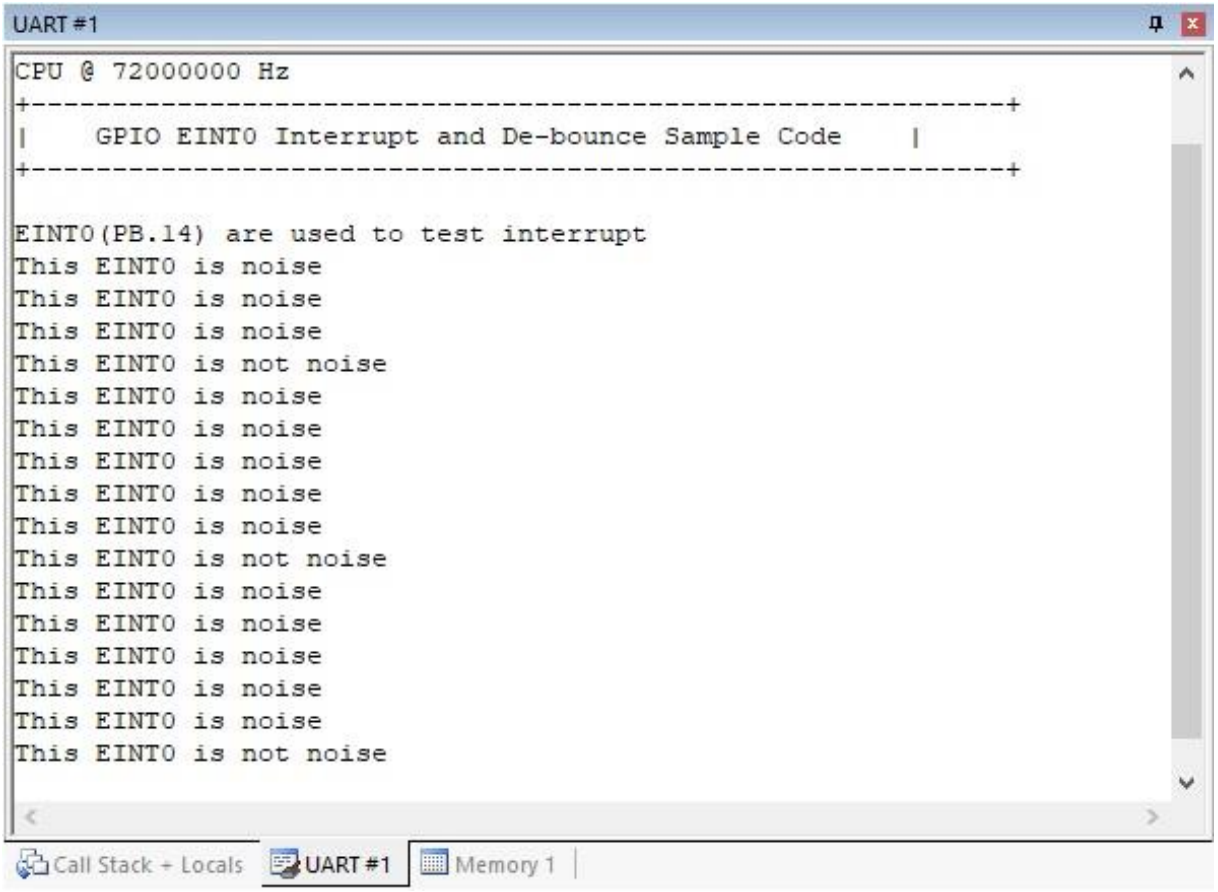
1.1 Introduction

Under normal circumstances, when the I/O detects the signal, it is likely to be disturbed by the noise, and the system will respond incorrectly. This example uses the system counter to filter the jitter that occurs during a certain time interval as noise to avoid making mistakes.

1.2 Principle

When the I/O detects the signal, enter the interrupt service routine and set the system counter (SysTick) in the interrupt service. The counter counts for a certain time interval, and the detected signal change during the time is regarded as noise. If no change is detected, it is a normal signal.

1.3 Demo Result



2 Code Description

EINT0 interrupt service routine, clear interrupt flag and set system counter, print out whether it is noise：

```
    void EINT0_IRQHandler(void){
         int i, deboundce=0;

         for(i=0;i<500;i++){
                  /* For PB.14, clear the INT flag */
                  GPIO_CLR_INT_FLAG(PB, BIT14);
                  SysTick->LOAD = 1000*CyclesPerUs;
                  SysTick->VAL  = 0;
                  SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;

                  /* Waiting for down-count to zero */
                  while((SysTick->CTRL & SysTick_CTRL_COUNTFLAG_Msk) == 0) {
                           if(PB14==1){
                                    PB13^=1;
                                    deboundce=1;
                                    printf("This EINT0 is noise\n");
                                    goto exit_int0;
                           }
                  }
         }

         if(deboundce==0){
                  PB12^=1;
                  printf("This EINT0 is not noise\n");
         }
    exit_int0:
         return;
    }
```

Set PB.14 to detect external signals and corresponding interrupt service programs：

```
/* Configure PB.14 as EINT0 pin and enable interrupt by falling edge trigger */
    GPIO_SetMode(PB, BIT14, GPIO_PMD_QUASI);
    GPIO_EnableEINT0(PB, 14, GPIO_INT_FALLING);
    NVIC_EnableIRQ(EINT0_IRQn);
```

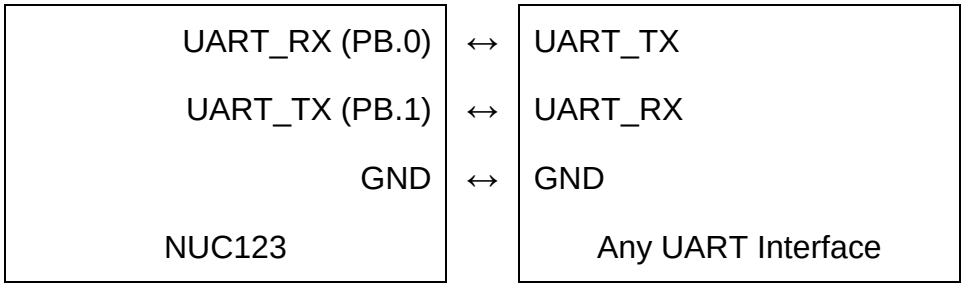
3 Software and Hardware Environment

Software Environment

- BSP version
  - NUC123 Series BSP CMSIS v3.01.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-123-LQFP64_V1.0
- Diagram



4 Directory Information

EC_NUC123_Software_Debounce_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC123_Software_Debounce.uvproj.
2. Enter KEIL compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct. 24, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.