Function Keys

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Perform the function key demo includes short pressed event, interval event and long pressed event
BSP Version	M480 Series BSP CMSIS v3.04.000
Hardware	NuMaker-PFM-M487 v3.0

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

This example code shows how to declare a function key which can identify short press event, interval event and long press event.

1.2 Principle

This example uses a GPIO interrupt to identify the status of key is pressed or released and a 1KHz timer to record the time interval between the pressed and released. And the short event, interval event and long press event can be identified according timer interval between pressed and released and status of key.

1.3 Demo Result

```
shortHandler
intervalHandler
intervalHandler
intervalHandler
intervalHandler
intervalHandler
intervalHandler
longHandler
```

2 Code Description

(1) Hook the initialize function and time tick function

(2) Setup the parameter of short period, interval period and interval delay

(3) Hook the callback function of short press event, interval event and long press event
Initialize the function key

(4) Call the function key handler in the main loop

```
/* SW2 function & parameter & callback function */
sw2.init = sw2_init;
sw2.getTick = sw2_getTick;

sw2.shortPeriod = 500;      // ms
sw2.intervalPeriod = 250;   // ms
sw2.intervalDelay = 1000;   // ms

sw2.shortHandler = sw2_shortHandler;
sw2.intervalHandler = sw2_intervalHandler;
sw2.longHandler = sw2_longHandler;

functionKey_init(&sw2);

while(1) {
    /* SW2 handler */
    functionKey_Handler(&sw2);
}
```

(1) Update the status of key when GPIO interrupt occurred.
(2) Clear the GPIO interrupt

```
void GPG_IRQHandler(void)
{
    /* To check if PG.15 interrupt occurred */
    if(GPIO_GET_INT_FLAG(PG, BIT15)) {
        GPIO_CLR_INT_FLAG(PG, BIT15);

        /* Record the key status */
        if (PG15 == 1) {
            functionKey_chgStatus(&sw2, RELEASED);
        } else {
            functionKey_chgStatus(&sw2, PRESSED);
        }

    } else {
        /* Un-expected interrupt. Just clear all PC interrupts */
        PG->INTSRC = PG->INTSRC;
        printf("Un-expected interrupts.\n");
    }
}
```

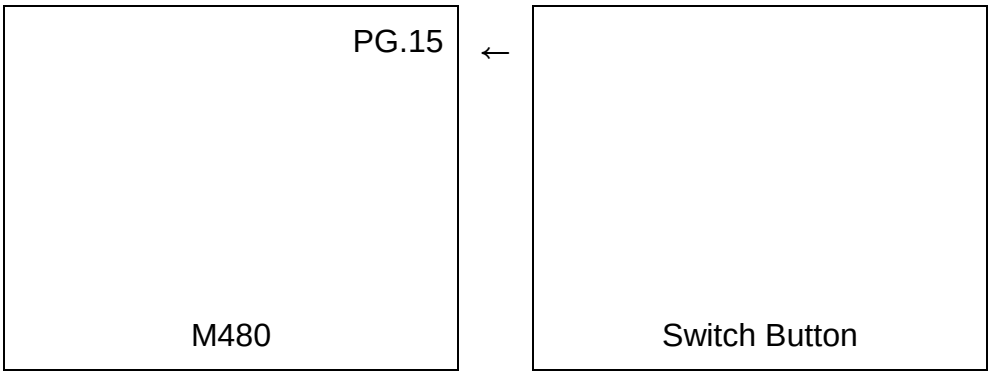
3 Hardware and Software environment

Software Environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-PFM-M487
- Diagram



4 Directory Information

- EC_M480_FunctionKeys_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
  - SamlpeCode
    - ExampleCode — Source file of example code

5 How to execute a sample code

1. Browsing into sample code folder by Directory Information (section 4) and double click M480_FunctionKeys.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

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.