

Measure VDD And Temperature

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Demonstrate measure internal temperature and supply voltage of IC using internal BandGap 2.5V voltage.
BSP Version	Nano100B Series BSP CMSIS V3.03.000
Hardware	NuTiny-EVB-Nano130-LQPF128

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

Demonstrate measure internal temperature and supply voltage of IC using internal BandGap 2.5V voltage.

1.2 Principle

Use internal BandGap 2.5V voltage as IC internal reference voltage.

Enable internal ADC channel 14 and enable temperature detection register to measure IC internal temperature.

The calculate function show as below:

$$V_{temp}(mV) = 2.5 * 1000 * (ADC_14 / 4096)$$

$$V_{temp} (mV) = -1.73 (mV/^\circ C) \times Temperature (^\circ C) + 740 (mV).$$

Measuring the supply voltage using ADC channel 15.

The calculate function show as below:

$$VDD = (4096 / ADC_15) * 2.5V$$

1.3 Demo Result

The screenshot shows a terminal window titled "COM3:115200baud - Tera Term VT". The output text is as follows:

```

CPU @ 31500000Hz

This sample code demonstrate measure VDD and Temperature.
It convert ADC channel 14, 15 and print conversion result
ADC channel 15 is VDD
ADC channel 14 is Temperature

Channel 15 conversion result is 3131, VDD is 3.269722 volt
Channel 14 conversion result is 891, Temperature is 17.612877 degree
    
```

2 Code Description

Enable temperature sensor and BandGap :

```
SYS_EnableTempCtl(); // Enable Temperature Sensor
SYS_EnableIntVRefBGP(1); // Enable Vref is BGP
```

Set ADC and start the A/D conversion :

```
// Set ADC
ADC_Open(ADC, ADC_INPUT_MODE_SINGLE_END, ADC_OPERATION_MODE_SINGLE_CYCLE,
ADC_CH_14_MASK | ADC_CH_15_MASK);
ADC_SetExtraSampleTime(ADC, ADC_CH_14_MASK, 1024);

ADC_SetVrefVoltage(0); // Set Vref voltage

// Power on ADC
ADC_POWER_ON(ADC);

// Enable ADC ADC_IF interrupt
ADC_EnableInt(ADC, ADC_ADF_INT);
NVIC_EnableIRQ(ADC_IRQn);

ADC_START_CONV(ADC);
```

Get A/D conversion value and calculate VDD voltage and temperature :

```
u32Result = ADC_GET_CONVERSION_DATA(ADC, 15);
fAvdd = 2.5 * 0xffff / u32Result;
printf(" Channel 15 conversion result is %d, VDD is %f volt \n", u32Result, fAvdd);
u32Result = ADC_GET_CONVERSION_DATA(ADC, 14);
fTmp = (740 - (1000 * u32Result / 4096) * fAvdd) / 1.73 ;
printf(" Channel 14 conversion result is %d, Temperature is %f degree \n", u32Result,
fTmp);
```

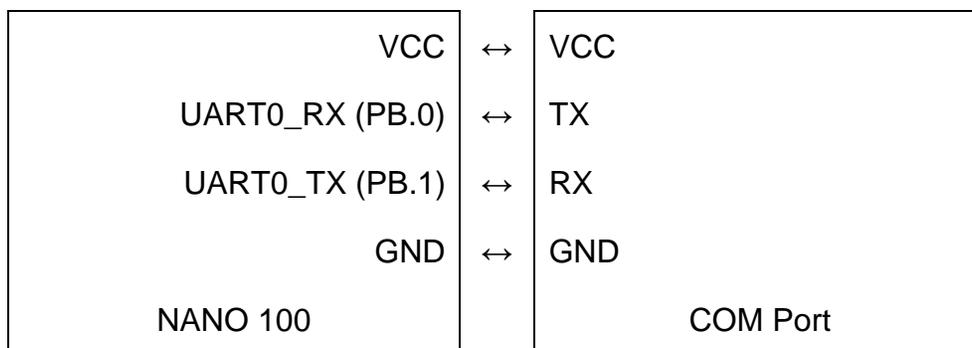
3 Software and Hardware Environment

- **Software Environment**

- BSP version
 - ◆ Nano100B Series BSP CMSIS V3.03.000
- IDE version
 - ◆ Keil uVersion 5.26

- **Hardware Environment**

- Circuit components
 - ◆ NuTiny-EVB-NANO130-LQPF128
- Diagram



4 Directory Information

- 
EC_Nano100_Use_BandGap_Measure_VDD_And_Temperature
 - 
Library
 - 
CMSIS
 - 
Device
 - 
StdDriver
 - 
SampleCode
 - 
ExampleCode

Sample code header and source files

Cortex[®] Microcontroller Software Interface Standard (CMSIS) by Arm[®] Corp.

CMSIS compliant device header file

All peripheral driver header and source files

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 Nano100_Use_BandGap_Measure_VDD_And_Temperature.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
Jun. 26, 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.*