How to use nano100B ADC ACMP

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Use ADC compare mode and use TestOScope to plot ADC data
BSP Version	Nano100B Series BSP CMSIS v3.03.000
Hardware	NuTiny-EVB-Nano130-LQFP128 v1.1

1 Function Description

1.1 Introduction

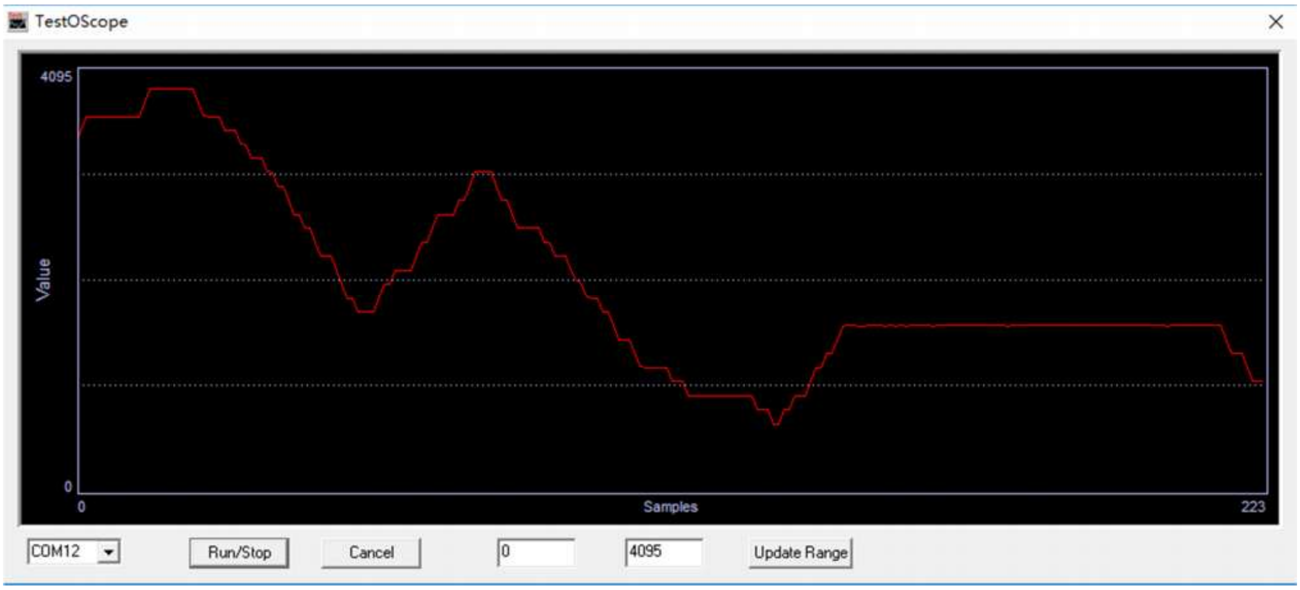
Sometimes we use the serial port to send the ADC data to the PC for analysis. TestOScope is the computer software, which will capture the serial port data to the user through the image, which is convenient for observing the trend of data changes.
We can also compare the value of the ADC with the built-in ACMP. When the comparison condition is met (such as the value of the ADC is greater than or equal to 2048 and meets 10 times), interrupt processing can be generated, such as buzzer or LED flashing.

1.2 Principle

The UART0 TX pin of MCU is PB.1 and the ADC input pin of MCU is PA.0.
When the serial port of the MCU is connected to the serial port of the PC, click the Run/Stop button of TestOScope, and the TestOScope will draw the graphic as long as the serial port of the MCU has data.
PF.5 will output 0, when the value of ADC matches the set comparison value, otherwise output 1.

1.3 Demo Result

The red line in the figure below is the trend of the data of the captured ADC data between 0 and 4095.



2 Code Description

2.1.1 Software ADC Initialization Setting

The difference from the normal ADC initialization procedure is that the ADC comparison function is enabled.

```
    void ADC_Init()
    {
        /*---------------------------------------------------------------------------------------------------*/
        /* Init ADC                                                                                         */
        /*---------------------------------------------------------------------------------------------------*/
        ADC_Open(ADC, ADC_INPUT_MODE_SINGLE_END, ADC_OPERATION_MODE_SINGLE,
ADC_CH_0_MASK);

        /* Set the reference voltage selection */
        ADC_SET_REF_VOLTAGE(ADC, ADC_REFSEL_POWER);

        /* Set the resolution of conversion result */
        ADC_SET_RESOLUTION(ADC, ADC_RESSEL_12_BIT);

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

        /* clear the A/D interrupt flag for safe */
        ADC_CLR_INT_FLAG(ADC, ADC_ADF_INT);

        /* Configure the comparator 0 and enable it */
        ADC_ENABLE_CMP0(ADC, 0, ADC_CMP_GREATER_OR_EQUAL_TO, 2048, 1);
        ADC_EnableInt(ADC, ADC_CMP0_INT);

        /* Power on ADC module */
        ADC_POWER_ON(ADC);
    }
```

2.1.2 Software ADC Interrupt handler

In addition to processing the ADC conversion completion in the ADC interrupt routine, the processing of ADC comparison results has been added.

```
void ADC_IRQHandler(void)
{
    uint32_t u32Flag;

    /* Get ADC conversion finish interrupt flag */
    u32Flag = ADC_GET_INT_FLAG(ADC, ADC_ADF_INT | ADC_CMP0_INT);

    if (u32Flag & ADC_ADF_INT)
    {
        g_u8ADF = 1;
        ADC_CLR_INT_FLAG(ADC, ADC_ADF_INT);
    }

    if (u32Flag & ADC_CMP0_INT)
    {
        ADC_CLR_INT_FLAG(ADC, ADC_CMP0_INT);
        PF5 = 0;
    }
    else
        PF5 = 1;
}
```

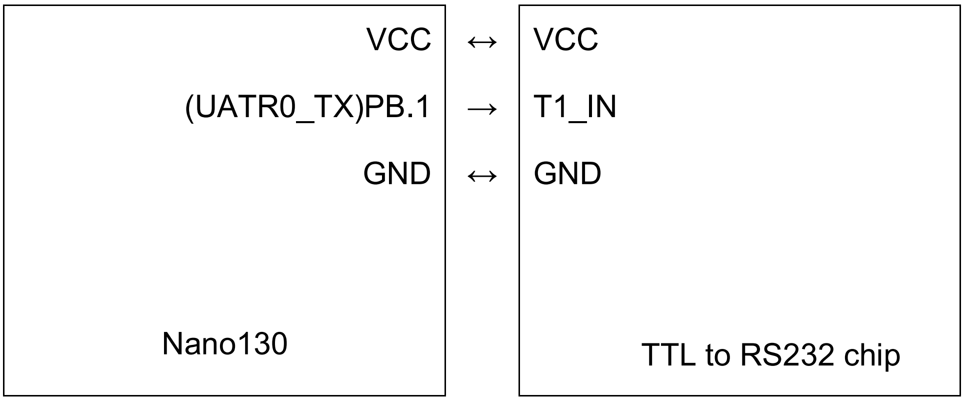
3 Software and Hardware Environment

Software Environment

- BSP version
  - Nano100B Series BSP CMSIS v3.03.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-EVB-Nano130-LQFP128 v1.1
  - TTL to RS232 chip
- Diagram



4 Directory Information

EC_Nano100B_ADC_RawData_Alarm_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 Nano100B_ADC_RawData_Alarm.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
Nov. 13, 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.