Use EADC to capture DAC signal

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses EADC to capture the sine wave produced by the DAC.
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-PFM-M487 Ver 3.0

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

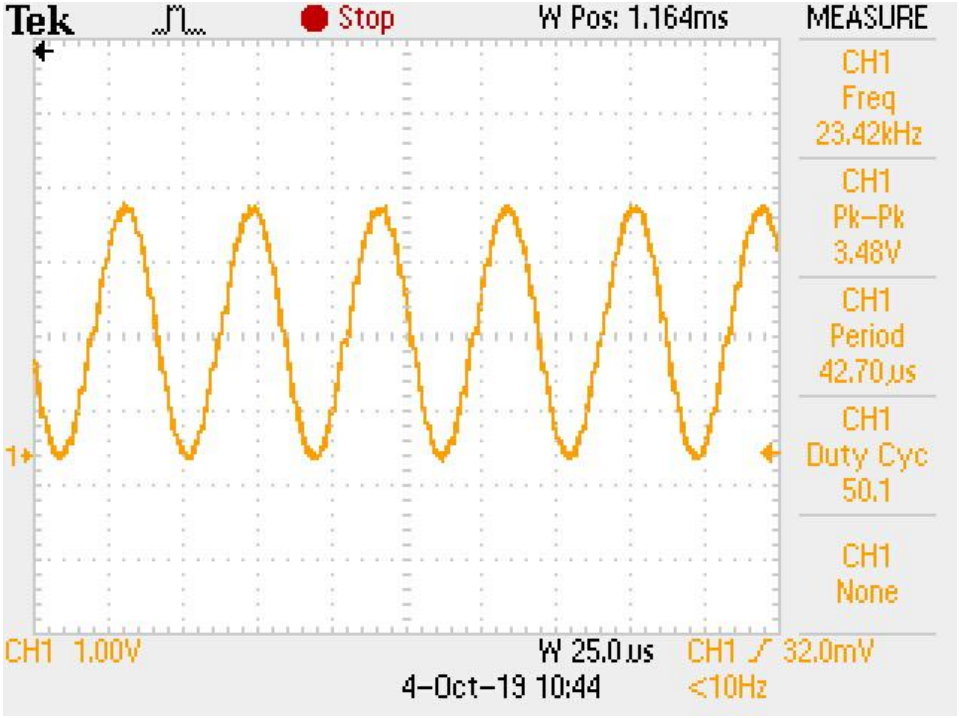
1 Function Description

1.1 Introduction

In the example code, the software triggers the DAC to output a sine wave, and after the DAC output is stable, the software triggers the EADC to read its value. The sine wave consists of 64 levels. The pin used by the DAC is PB.13, and the pin used by the EADC is PB.0. The user must connect the two pins.

1.2 Demo Result

From the EADC to the signal as shown below



2 Code Description

First initialize the DAC and EADC modules as software triggers:

```
void DAC_Init()
{
    /* Set the software trigger DAC and enable D/A converter */
    DAC_Open(DAC1, 0, DAC_SOFTWARE_TRIGGER);

    /* The DAC conversion settling time is 1us */
    DAC_SetDelayTime(DAC1, 1);

    /* Set DAC 12-bit holding data */
    DAC_WRITE_DATA(DAC1, 0, 0x400);
}
void EADC_Init()
{
    /* Set input mode as single-end and enable the A/D converter */
    EADC_Open(EADC, EADC_CTL_DIFFEN_SINGLE_END);

    /* Configure the sample module 0 for analog input channel 2 and software trigger
source.*/
    EADC_ConfigSampleModule(EADC, 0, EADC_SOFTWARE_TRIGGER, 0);

    /* Clear the A/D ADINT0 interrupt flag for safe */
    EADC_CLR_INT_FLAG(EADC, EADC_STATUS2_ADIF0_Msk);
}
```

The software then triggers the DAC in the while loop and waits for the DAC output to complete before the software triggers the EADC to begin reading data.

```
while(1) {
        for(index = 0; index < 64; index++) {
            /* Write data to DAC */
            DAC_WRITE_DATA(DAC1, 0, sine[index++]);
            /* Software trigger DAC to start converting */
            DAC_START_CONV(DAC1);
            /* Wait DAC convertion complete */
            while(DAC_IS_BUSY(DAC1,0));

            /* Software trigger ADC to start converting */
```

```
            EADC_START_CONV(EADC, BIT0);
            /* Wait ADC convertion complete */
            while(EADC_IS_BUSY(EADC));
            /* Get the conversion result of the sample module 0 */
            i32ConversionData = EADC_GET_CONV_DATA(EADC, 0);
        }
    }
```

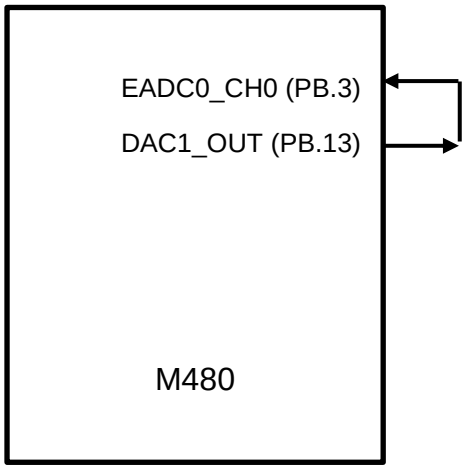
3 Software and Hardware Environment

Software environment

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

Hardware environment

- Circuit components
  - NuMaker-IoT-M487 or other M480 Development Board
- Diagram



4 Directory Information

- EC_M480_EADCCaptureDAC_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 EADCCaptureDAC.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. 1, 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.