Subsystem Design

Simultaneous Sampling of ADCs

1 Description

This subsystem demonstrates how to simultaneously sample signals across 2 ADC instances by utilizing the event fabric of the MSPM0. This process reduces time skew and keeps the signals in phase, simplifying calculations for applications such as E-metering, motor control, or medical signal processing.

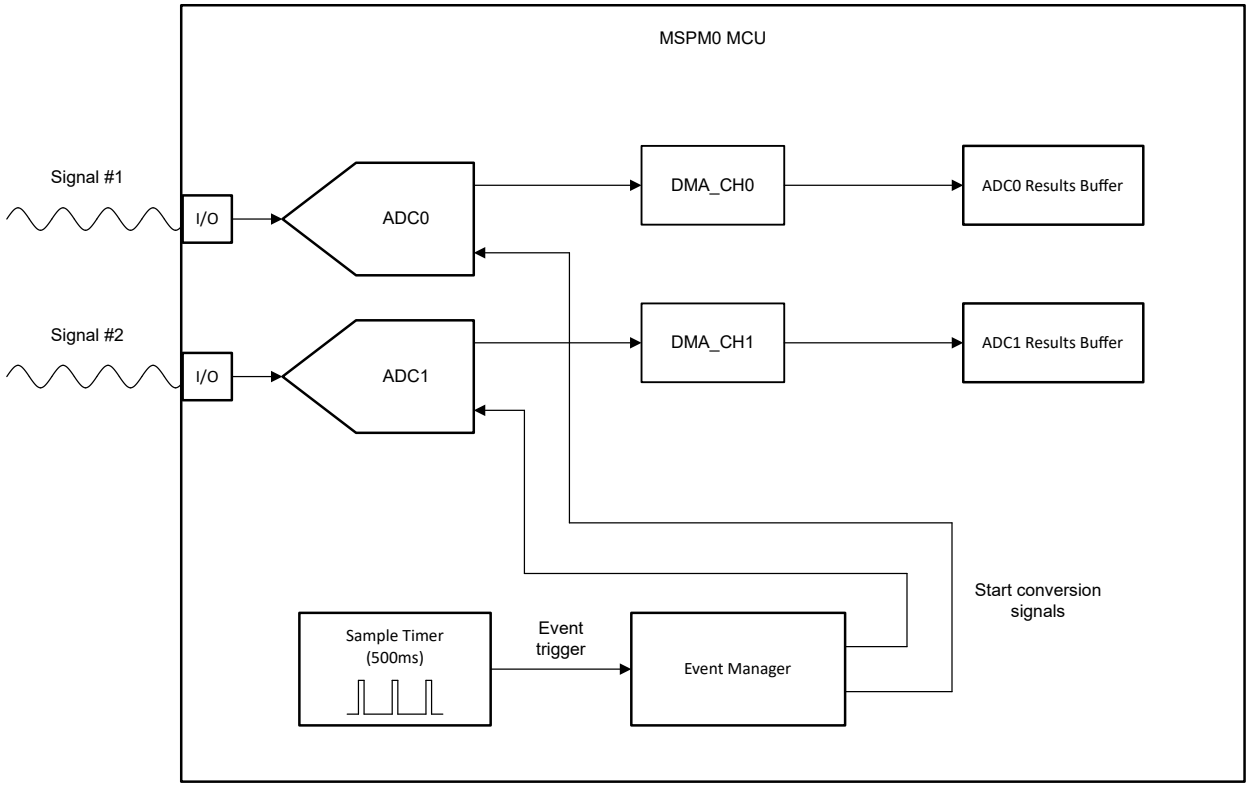

Figure 1-1. Simultaneous Sampling Functional Block Diagram

2 Required Peripherals

This application requires two integrated ADCs, a DMA module with at least two channels, and a TIM module.

Table 2-1. Required Peripherals

Sub-block Functionality	Peripheral Use	Notes
Analog Signal Capture	(2×) ADC	Shown as *ADC12_0_INST* and *ADC12_1_INST* in code
Memory Transfer	(1×) DMA	Shown as *DMA* in code
Event Timer	(1x) TIM	Shown as *TIMER_0_INST* in code

3 Design Steps

1. Determine the configuration for the ADCs including reference source, reference value, resolution, and sampling rate based on the given analog input and design requirements.
2. Determine the period of the timer that triggers the ADCs based on design requirements.
3. Generate two array buffers with sizes matching those of the DMA transfer size to store all of the ADC conversion results.
4. In SysConfig, configure the timer as an event publisher to a 1:2 channel and have each ADC configured as a subscriber to the same channel.
5. Write *Application Code* to start the timer and process the simultaneously sampled data.

4 Design Considerations

1. *Maximum Sampling Speed:* The sampling speed of the ADC is based on input signal frequency, analog front end, filters, or any other design parameters that affect sampling.
2. *ADC Reference:* Choose the reference to align with the expected maximum input to utilize the full scale range of the ADC.
3. *Clock Settings:* The clock source determines the total time for the conversion. The clock divider in tandem with the SCOMP setting determines the total sampling time. SysConfig sets the appropriate SCOMP depending on the sampling time setting.
4. *Timer Period:* The timer period is based on how frequently the system needs to sample the input signals. Different combinations of Timer Clock Divider and Timer Clock Prescaler in SysConfig can provide the desired resolution.

5 Software Flowchart

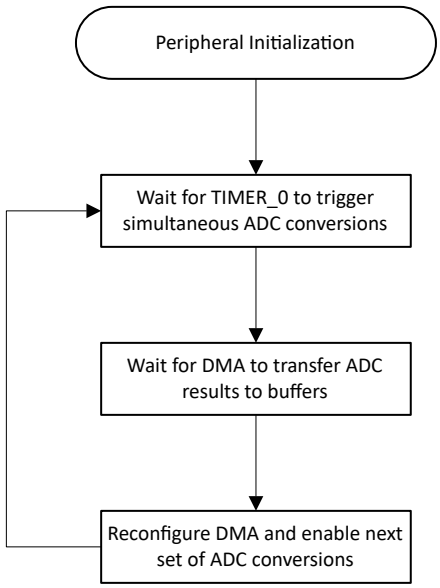

Figure 5-1. Example Software Sequence

6 Application Code

```
#include"ti_msp_dl_config.h"

#define ADC_SAMPLE_SIZE (64)

/* When FIFO is enabled, two 12bit samples are compacted into a single 32bit word */
#define ADC_FIFO_SAMPLES (ADC_SAMPLE_SIZE/2)
uint16_t gADC0Samples[ADC_SAMPLE_SIZE];
uint16_t gADC1Samples[ADC_SAMPLE_SIZE];

volatile bool gADC0Done;
volatile bool gADC1Done;

int main(void) {
   SYSCFG_DL_init();
   /* Configure DMA source, destination and size */
   DL_DMA_setSrcAddr(DMA, DMA_CH0_CHAN_ID, (uint32_t)DL_ADC12_getFIFOAddress(ADC12_0_INST));
   DL_DMA_setDestAddr(DMA, DMA_CH0_CHAN_ID, (uint32_t)&gADC0Samples[0]);
   DL_DMA_setTransferSize(DMA, DMA_CH0_CHAN_ID, ADC_FIFO_SAMPLES);
   DL_DMA_enableChannel(DMA, DMA_CH0_CHAN_ID);

   DL_DMA_setSrcAddr(DMA, DMA_CH1_CHAN_ID, (uint32_t)DL_ADC12_getFIFOAddress(ADC12_1_INST));
   DL_DMA_setDestAddr(DMA, DMA_CH1_CHAN_ID, (uint32_t)&gADC1Samples[0]);
   DL_DMA_setTransferSize(DMA, DMA_CH1_CHAN_ID, ADC_FIFO_SAMPLES);
   DL_DMA_enableChannel(DMA, DMA_CH1_CHAN_ID);

   /* Enable interrupts for both ADCs */
   NVIC_EnableIRQ(ADC12_0_INST_INT_IRQN);
   NVIC_EnableIRQ(ADC12_1_INST_INT_IRQN);
   gADC0Done = false;
   gADC1Done = false;

   /* Start timer */
   DL_TimerG_startCounter(TIMER_0_INST);
   while (1) {
      while (gADC0Done == false || gADC1Done == false) {
         __WFI();
      }

      /* Breakpoint to check the buffers */
      __BKPT(0);

      /* Reconfigure DMA */
      DL_DMA_setSrcAddr(DMA, DMA_CH0_CHAN_ID, (uint32_t)DL_ADC12_getFIFOAddress(ADC12_0_INST));
      DL_DMA_setDestAddr(DMA, DMA_CH0_CHAN_ID, (uint32_t)&gADC0Samples[0]);
      DL_DMA_setTransferSize(DMA, DMA_CH0_CHAN_ID, ADC_FIFO_SAMPLES);
      DL_DMA_enableChannel(DMA, DMA_CH0_CHAN_ID);

      DL_DMA_setSrcAddr(DMA, DMA_CH1_CHAN_ID, (uint32_t)DL_ADC12_getFIFOAddress(ADC12_1_INST));
      DL_DMA_setDestAddr(DMA, DMA_CH1_CHAN_ID, (uint32_t)&gADC1Samples[0]);
      DL_DMA_setTransferSize(DMA, DMA_CH1_CHAN_ID, ADC_FIFO_SAMPLES);
      DL_DMA_enableChannel(DMA, DMA_CH1_CHAN_ID);
      break;

      gADC0Done = false;
      gADC1Done = false;
   }
}

void ADC12_0_INST_IRQHandler(void) {
   switch (DL_ADC12_getPendingInterrupt(ADC12_0_INST)) {
   case DL_ADC12_IIDX_DMA_DONE:
      gADC0Done = true;
      DL_ADC12_enableConversions(ADC12_0_INST);
      break;
   default:
      break;
   }
}

void ADC12_1_INST_IRQHandler(void) {
   switch (DL_ADC12_getPendingInterrupt(ADC12_1_INST)) {
   case DL_ADC12_IIDX_DMA_DONE:
      gADC1Done = true;
      DL_ADC12_enableConversions(ADC12_1_INST);
   default:
      break;
```

```
    }
}
```

7 Additional Resources

- Texas Instruments, *MSPM0 G-Series 80-MHz Microcontrollers*, technical reference manual.
- Texas Instruments, *MSPM0G350x Mixed-Signal Microcontrollers with CAN-FD Interface*, data sheet.

8 E2E

See TI's E2E™ support forums to view discussions and post new threads to get technical support for using MSPM0 devices in designs.

9 Trademarks

E2E™ is a trademark of Texas Instruments.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265