

1 Description

This subsystem demonstrates how the internal ADC, and math accelerator (MATHACL) modules within the MSPM0G family of devices can be used to implement a simple, streaming IIR filter of an analog signal. In this configuration, noise on an analog signal is filtered using a single pole IIR filter. The defined beta value can be adjusted to control the IIR filter decay over frequency.

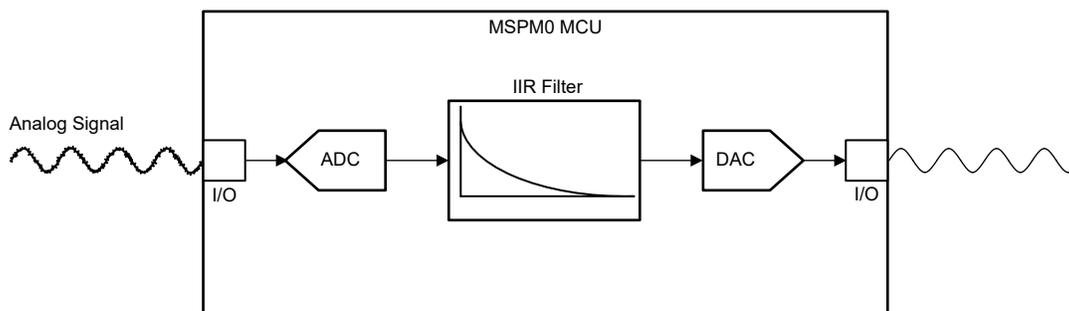


Figure 1-1. IIR Filter Functional Block Diagram

2 Required Peripherals

Required Peripherals

This application requires an integrated ADC, MathACL, and DAC12 modules.

Table 2-1. Required Peripherals

Sub-block Functionality	Peripheral Use	Notes
Analog Signal Capture	(1×) ADC	Shown as <code>ADC12_0_INST</code> in code
IIR Filter	(1×) MathACL	Shown as <code>MATHACL</code> in code
Analog Signal Output (Optional)	(1×) DAC12	Shown as <code>DAC12_0_INST</code> in code

3 Compatible Devices

Based on the requirements in [Table 2-1](#), this example is compatible with the devices listed in [Table 3-1](#). The corresponding EVM can be used for prototyping.

Table 3-1. Compatible Devices

Compatible Devices	EVM
MSPM0G35xx, MSPM0G15xx	LP-MSPM0G3507

4 Design Steps

1. Determine the minimum required ADC sampling frequency. This must be at least twice the bandwidth of the input signal.
2. Determine the desired rejection coefficient. The rejection coefficients in a single pole IIR filter governs the rate of decay of the filter over frequency. The rejection coefficient is sometimes referred to as the beta (β) value, or the decay value.
 - a. There are different tools for IIR filter coefficient calculation, which is not discussed in this document.
3. Convert the filter coefficient to a fixed point value.
 - a. In the example code, a Q8 (eight fractional bits) representation is used. Perform this conversion using the [IQMath library](#) or by multiplying the coefficients by 2^n where n is the desired number of fractional bits. Verify that the selected data type can hold these values without overflowing.
 - b. The filter coefficients are constant values and can be contained in flash to save room in SRAM if desired.

5 Design Considerations

1. **Input signal bandwidth:**
The bandwidth of the signal that must be resolved determines the ADC sampling frequency and the amount of data the code must process.
2. **ADC reference voltage:**
The ADC reference voltage must be selected such that the signal amplitude can be fully captured with good resolution.
3. **Decay coefficient:**
In a single-pole IIR filter, the decay value is the single coefficient that weights the contribution of new samples to the current result. The magnitude of the decay coefficient is between zero and one. A higher decay value results in an earlier cutoff frequency.

6 Software Flow Chart

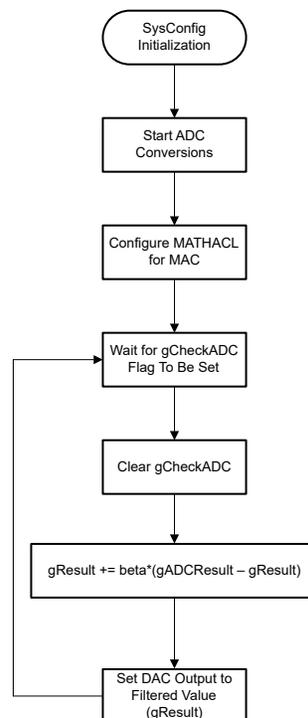


Figure 6-1. Example Software Sequence

7 Application Code

```

volatile bool gCheckADC;
/* Filtered Result */
uint32_t gResult = 0;
/* ADC Value Output */
uint32_t gADCResult = 0;

/* Scaling Factor, Q8 value (0-255) */
uint32_t gBeta = 16;
const DL_MathACL_operationConfig gMpyConfig = {
    .opType      = DL_MATHACL_OP_TYPE_MAC,
    .opSign      = DL_MATHACL_OPSIGN_SIGNED,
    .iterations  = 0,
    .scaleFactor = 0,
    .qType       = DL_MATHACL_Q_TYPE_Q8};
int main(void)
{
    SYSCFG_DL_init();
    NVIC_EnableIRQ(ADC12_0_INST_INT_IRQN);
    gCheckADC = false;
    DL_ADC12_startConversion(ADC12_0_INST);

    /* Configure MathACL for Multiply and Accumulate */
    DL_MathACL_configOperation(MATHACL, &gMpyConfig, 0, 0 );
    DL_MathACL_enableSaturation(MATHACL);

    while (1) {
        while (false == gCheckADC) {
            __WFE();
        }
        gCheckADC = false;

        /* Calculate IIR Filter Output */
        gADCResult = DL_ADC12_getMemResult(ADC12_0_INST, DL_ADC12_MEM_IDX_0);
        /* Set Operand One last */
        DL_MathACL_setOperandTwo(MATHACL, gADCResult - gResult);
        DL_MathACL_setOperandOne(MATHACL, gBeta);
        DL_MathACL_waitForOperation(MATHACL);
        gResult = DL_MathACL_getResultOne(MATHACL);
        DL_DAC12_output12(DAC0, gResult);
    }
}
/* Set the ADC Result flag to trigger our main loop to process the new data */
void ADC12_0_INST_IRQHandler(void)
{
    switch (DL_ADC12_getPendingInterrupt(ADC12_0_INST)) {
        case DL_ADC12_IIDX_MEM0_RESULT_LOADED:
            gCheckADC = true;
            break;
        default:
            break;
    }
}

```

8 Additional Resources

- Texas Instruments, [MSPM0 G-Series 80-MHz Microcontrollers](#), technical reference manual.
- Texas Instruments, [MSPM0 L-Series 32-MHz Microcontrollers](#), technical reference manual.
- Texas Instruments, [MSPM0G350x Mixed-Signal Microcontrollers With CAN-FD Interface](#), data sheet.
- Texas Instruments, [MSPM0G150x Mixed-Signal Microcontrollers](#), data sheet.
- Texas Instruments, [MSPM0L130x Mixed-Signal Microcontrollers](#), data sheet.

9 E2E

See TI's [E2E](#) support forums to view discussions and post new threads to get technical support for using MSPM0 devices in designs.

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](https://www.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
Copyright © 2024, Texas Instruments Incorporated