Subsystem Design

High Gain Self-Compensating ADC

1 Description

This subsystem uses two cascaded op amps with DAC-controlled biasing to remove DC drift and prevent output saturation. A feedback loop centers the signal in real time, preserving dynamic range, and enabling UART transmission.

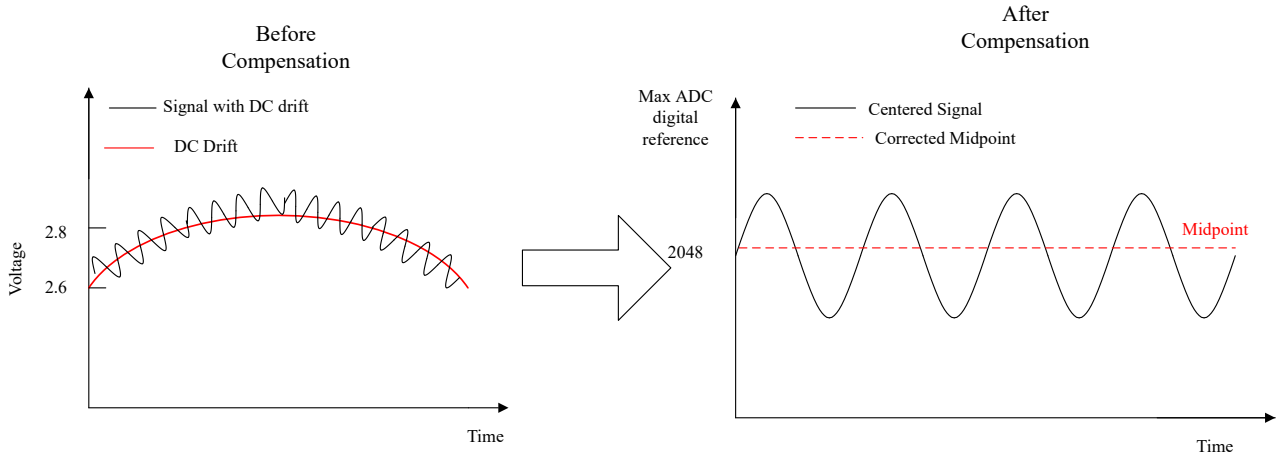

Figure 1-1. 1 Real-Time Correction of DC Drift

The subsystem consists of a cascaded two-stage op-amp architecture implemented using the MSPM0's internal OPAs, ADCs, and DACs. The first stage (OPA0) amplifies the input signal with a programmable gain (Gain 1) and applies a DC offset through DAC8_0 to maintain the signal is centered around the ADC midpoint. The second stage (OPA1), also with programmable gain (Gain 2), provides further amplification while the output is monitored and dynamically corrected using DAC8_1 to avoid saturation. Both stages are actively controlled in real-time through ADC feedback and digital compensation logic to maintain good dynamic range for signal acquisition.

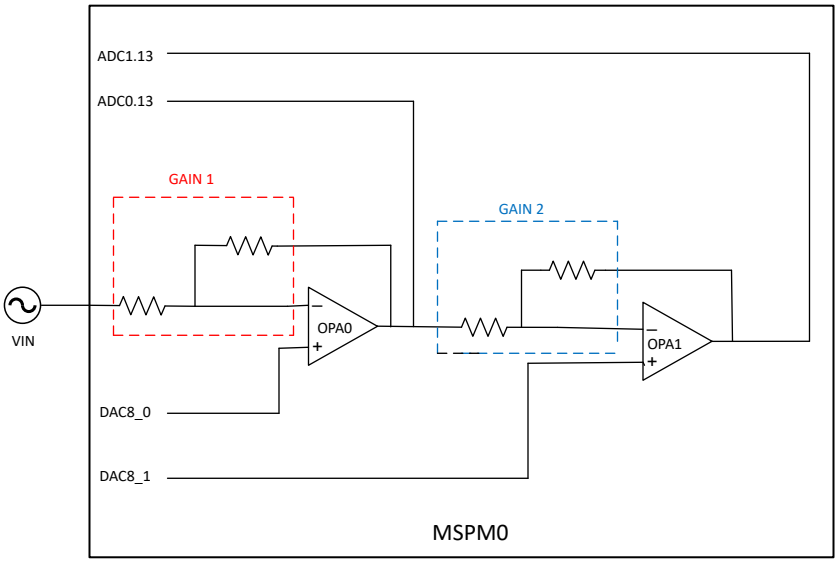

Figure 1-2. High Gain Self-Compensating ADC Architecture

2 Required Peripherals

Table 2-1 describes the required integrated peripherals.

Table 2-1. Required Peripherals

Sub-block	Peripheral Used	Notes
First-stage DAC bias control	DAC8_0 (via COMP_0)	Adjusted based on EMA of amplified input
Second-stage DAC saturation control	DAC8_1 (via COMP_1)	Regulates output to prevent saturation
Input/output voltage measurement	ADC12	Monitors OPA0 and OPA1 outputs
Timing control	TIMERX	Sets the system sampling rate
Data transmission	UART	Sends cascade output over UART as 4-byte packets

3 Design Steps

1. Configure two op amps with internal DACs and two ADC channels. Set up a UART interface for data output and a timer to control the sampling rate.
2. Power up the system and initialize DAC outputs to bias both op amp stages into valid operating ranges.
3. Define system-specific parameters such as amplifier gains, ADC thresholds, smoothing factor, and packet size in the customizable features header file.
4. Measure the first-stage output voltage, estimate the DC component using an EMA filter, and adjust the DAC to center the signal around the ADC midpoint.
5. Use the outputs from both stages to compute the final cascaded signal corresponding to the amplified input.
6. Monitor the second-stage output and adjust the DAC bias to keep the signal within range and prevent clipping or saturation.
7. Package the final amplified value into a 4-byte frame and send over UART for external logging or analysis.

4 Design Considerations

1. *Dynamic Range Centering:* The op amp outputs must remain centered within the ADC's input range to maximize resolution and avoid clipping, especially when DC drift is present.
2. *Gain Selection and Stability:* Proper gain settings for each stage are crucial to maintain the input signal is amplified enough for accurate detection without pushing the signal into saturation.
3. *EMA Filter Responsiveness:* The exponential moving average filter must be tuned to track slow DC drift effectively while ignoring the AC signal and high-frequency noise.

4. *Sampling Rate Control:* The timer must be configured to regulate how often signal measurements occur. The rate must be fast enough to track drift in real time, yet slow enough to avoid reacting to high-frequency content or causing unnecessary DAC updates.

5 Software Flow Chart

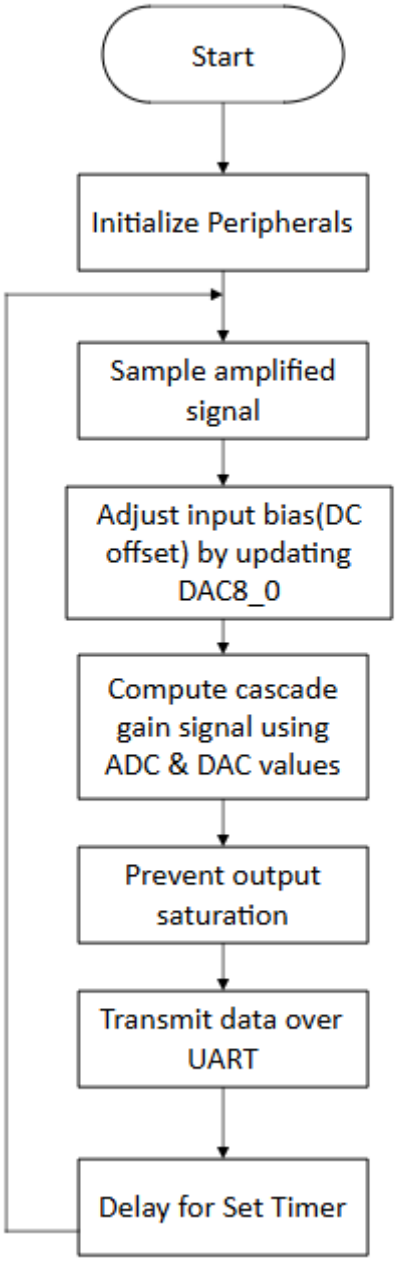

Figure 5-1. Software Flow Chart

6 Device Configuration

This subsystem utilizes TI's SysConfig graphical tool to configure all necessary peripherals. As shown in Figure Figure 6-1, the MSPM0 device internally routes the input signal through two cascaded op amp stages (OPA0 and OPA1), each configured as an inverting amplifier with programmable gain using internal resistor ladder taps (RTOP and RBOT). DAC8_0 and DAC8_1 provide biasing voltages to the non-inverting inputs of OPA0 and OPA1 respectively, enabling dynamic DC offset control at each stage.

The outputs of the op amps are monitored by ADC channels ADC0.13 and ADC1.13. TIMERX is configured to generate periodic interrupts that regulate the signal sampling rate, maintaining stable compensation over time. UART is also configured for serial communication over the backchannel interface, transmitting the cascaded output as a fixed 4-byte data packet.

All critical parameters—such as gain values, saturation limits, and filter constants—are defined in customizable_features.h to allow easy tuning without altering the core application logic.

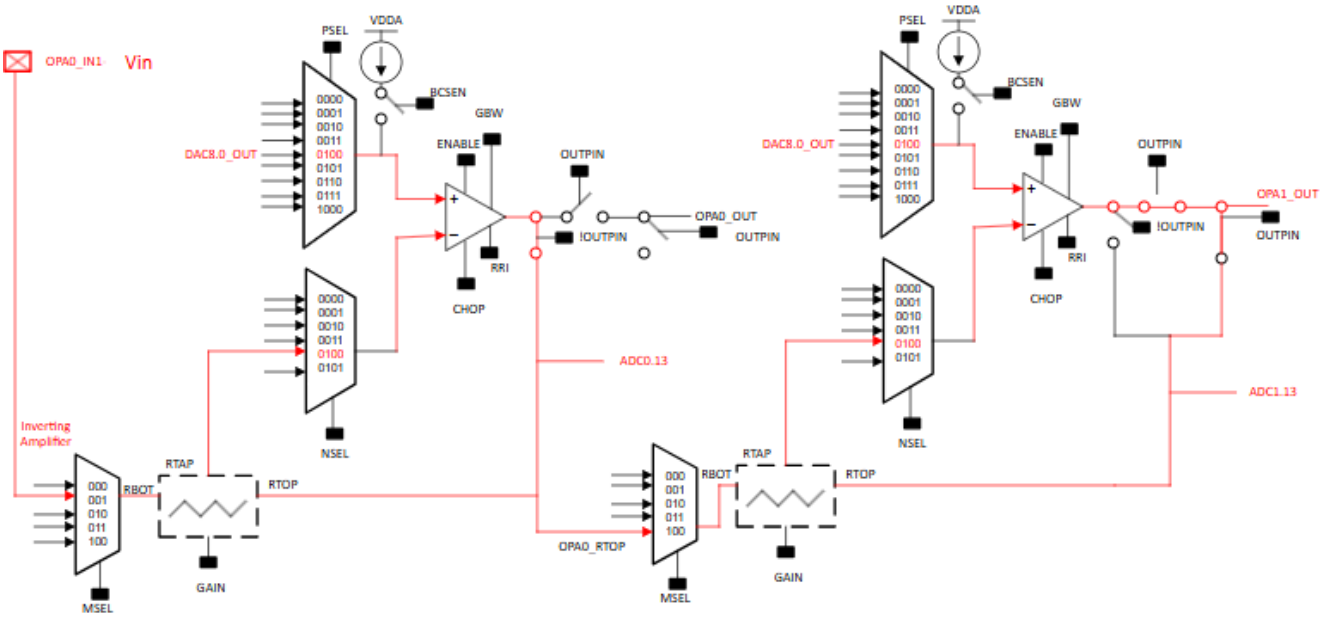

Figure 6-1. Inverting Cascade Amplifier Configuration

7 Application Code

The main loop runs a self-compensating sequence that includes DC bias adjustment, cascaded voltage calculation, saturation control, and UART transmission.

```
int main(void){
/**** Beginning of hardware initialization *********************************/

    // Initialize Hardware
    SYSCFG_DL_init();

    //Initialize interrupts and exceptions via the NVIC.
    NVIC_EnableIRQ(TIMER_1_INST_INT_IRQN);
    NVIC_EnableIRQ(ADC12_0_INST_INT_IRQN);
    NVIC_EnableIRQ(ADC12_1_INST_INT_IRQN);

/**** End of hardware initialization ***************************************/

/**** Beginning of software initialization *********************************/

    // Initialize DACs
    dacs_init();
/**** End of software initialization ***************************************/

    //Start TimerA
    DL_TimerA_startCounter(TIMER_1_INST);

    while(1){
        //Self-Compensating Functions
        remove_dc_voltage();                                        //Adjusts DAC8_0 based on input DC
        cascade_gained_input_voltage = cascade_input_voltage(OPA1_output);      //Calculates Vin x G1 x G2
        saturation_control(OPA1_output);                            //Adjusts DAC8_1 based on OPA1 output

        //Transmit Cascade Gained Data over UART
        send_UART_Data();

        //Increment pointer but not looping
        Data_pointer += 1;
        if(Data_pointer == DATALOGGING_LENGTH){Data_pointer = DATALOGGING_LENGTH-1;}

        //TIMERA_1 time delay for sampling interval:  50ms            //Configurable in SysConfig
        while (false == TIMERA_1_Timeout) {
            __WFI();
        }
        TIMERA_1_Timeout = false;
    }
}
```

Figure 7-1. Main.c code

- **remove_dc_voltage()** calculates the amplified input using the output of the first op amp and DAC reference. An exponential moving average (EMA) filter isolates the DC component, and the DAC is adjusted to center the output voltage around mid-scale.
- **cascade_input_voltage()** computes the cascaded signal G1·G2·Vin using both DAC values and op amp outputs. This output represents the system's filtered and gain-corrected signal.
- **saturation_control()** prevents output clipping in the second op amp by adjusting the DAC reference if the output voltage exceeds defined limits.
- **send_UART_Data()** breaks the 32-bit cascaded output into 4 bytes and transmits over UART.
- Timer and ADC interrupts control periodic execution and sampling. The TIMERA ISR sets a timeout flag every 50 ms, while ADC ISRs capture and store conversion results.

All tunable system constants such as gain values, voltage thresholds, buffer length, and smoothing factor are defined in customizable_features.h.

7.1 Application Code – Cascaded Signal Computation

A key part of this subsystem's functionality is computing the fully cascaded signal output, represented as G1 × G2 × V_{in}. This is handled in the cascade_input_voltage() function, which reconstructs the total amplified input by accounting for both op amp stages and the bias voltages applied through DACs.

```
/**** Function ************************************************************/

/*! \brief Calculates Cascaded Gained Input:  G1*G2*Vin1
 *
 *  (G1*G2*Vin1) = Vout2 + ((1+G1)*G2*Vdac1)*2^4 - ((1+G2)*Vdac2*2^4)
 *
 *  \return G1*G2*Vin1
 */
uint32_t cascade_input_voltage (uint32_t gOPA1_output)
{
    uint32_t result;
    /*
     * (G1*G2*Vin1/4096)*VDDA = (Vout2/4096)*VDDA + ((1+G1)*G2*Vdac1/256)*VDDA - ((1+G2)*Vdac2/256)*VDDA
     * (G1*G2*Vin1) = Vout2 + ((1+G1)*G2*Vdac1)*2^4 - ((1+G2)*Vdac2*2^4)
     *
     * for:
     *      G1 = OPA0_Gain
     *      G2 = OPA1_Gain
     */
    result = (gDAC8_0_DATA * 4) * ((OPA0_Gain+1) * OPA1_Gain );
    result -= (gDAC8_1_DATA * 4) * (OPA1_Gain + 1);
    result +=  gOPA1_output;

    return (result);
}   /* calc_cascade_input_voltage() */
```

Figure 7-2. Cascade_Input_Voltage() Code

The mathematical model behind the computation is:

$$\left(G1 \times G2 \times V_{in}1\right) = V_{out}2 + \left(\left(1 + G1\right) \times G2 \times V_{dac}1\right) \times 2^4 - \left(\left(1 + G2\right) \times V_{dac}2 \times 2^4 \quad (1)$$

Derived from the combination of

$$V_{out}1 = -G1 \times V_{in}1 + (1 + G1) \times V_{dac}1 \quad (2)$$

$$V_{out}2 = -G2 \times V_{out}1 + (1 + G2) \times V_{dac}2 \quad (3)$$

Where:

- G1 and G2 are the gains of the first and second op amp stages.
- Vout1 and Vout2 are the outputs of the first and second-stage amplifiers (OPA0 and OPA1).
- Vdac1 and Vdac2 are the bias voltages applied to OPA0 and OPA1 respectively through DAC8_0 and DAC8_1.

This equation compensates for the effects of both DAC-induced offsets in each stage and reconstructs the true input signal as if no offset biasing had occurred. The use of a left shift (<< 4) in the code accounts for the 8-bit DAC's limited resolution when scaled back to a 12-bit context, as each DAC value must be multiplied by 16 (for example, 2^4) to align with the full ADC range. This reconstructed signal is then transmitted via UART, providing the host system with a DC-compensated representation of the original input voltage, scaled by the configured system gains

8 Results

Figure 8-1 shows measured outputs from the two-stage amplifier: Orange (C1) as the output of OPA0, and Blue(C2) as the output of OPA1. The input signal is a 250mHz, 10mVpp sine wave with a 505mV DC offset. As seen, OPA1 further amplifies the OPA0 output, but due to high gain, limited supply headroom, and saturation accountability, the waveform does not linearly follow the input unlike OPA0 output that remains relatively smooth, while experiencing a lower gain and thus operates well within the linear range.

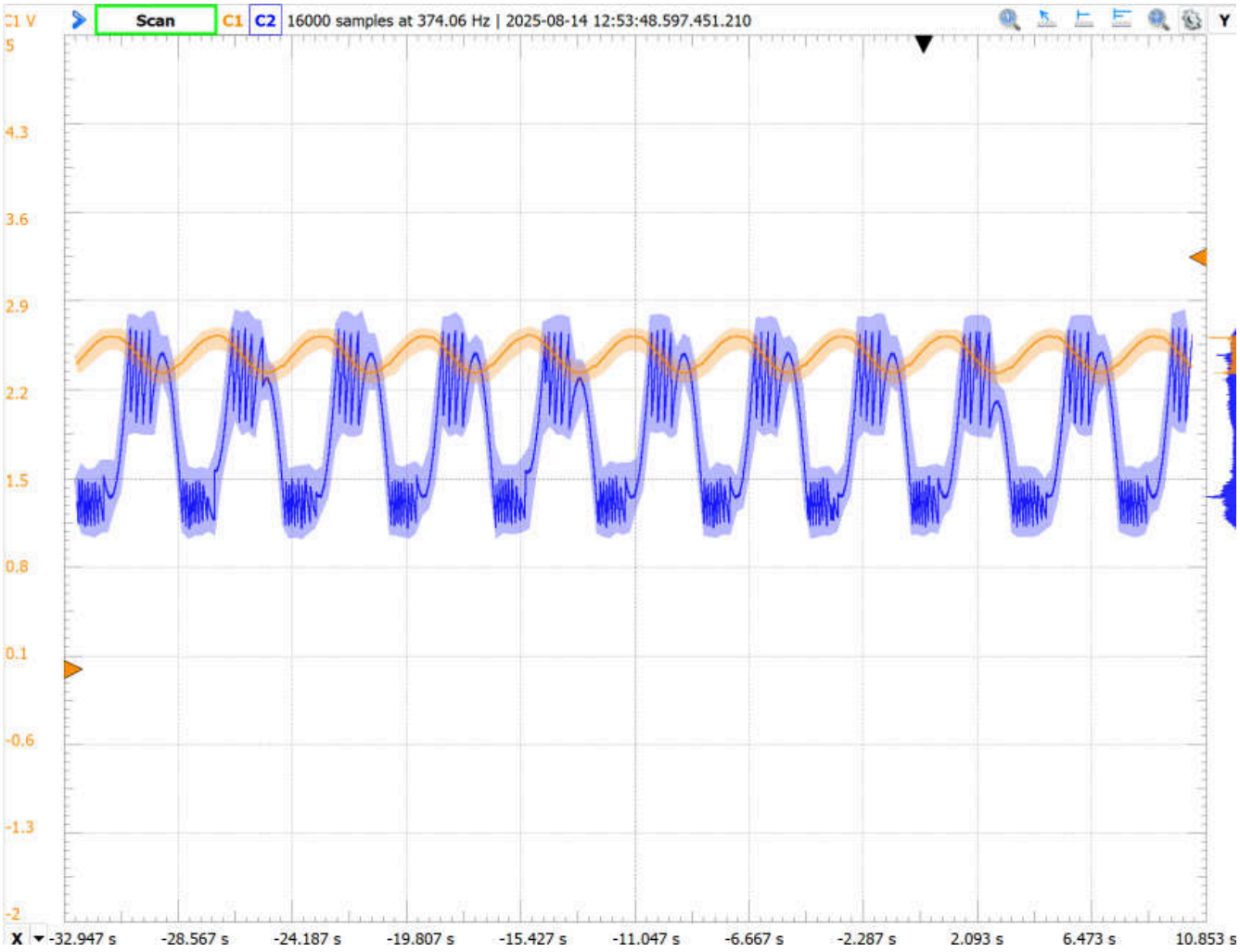

Figure 8-1. OPA0 and OPA1 Outputs

To address this, the system, as stated in the previous section, calculate the true amplified signal (G1 × G2 × V_{in}) in software, reconstructing the undistorted version based on known gain and bias settings, even if the physical output clips. Figure 8-2 represents the cascade signal.

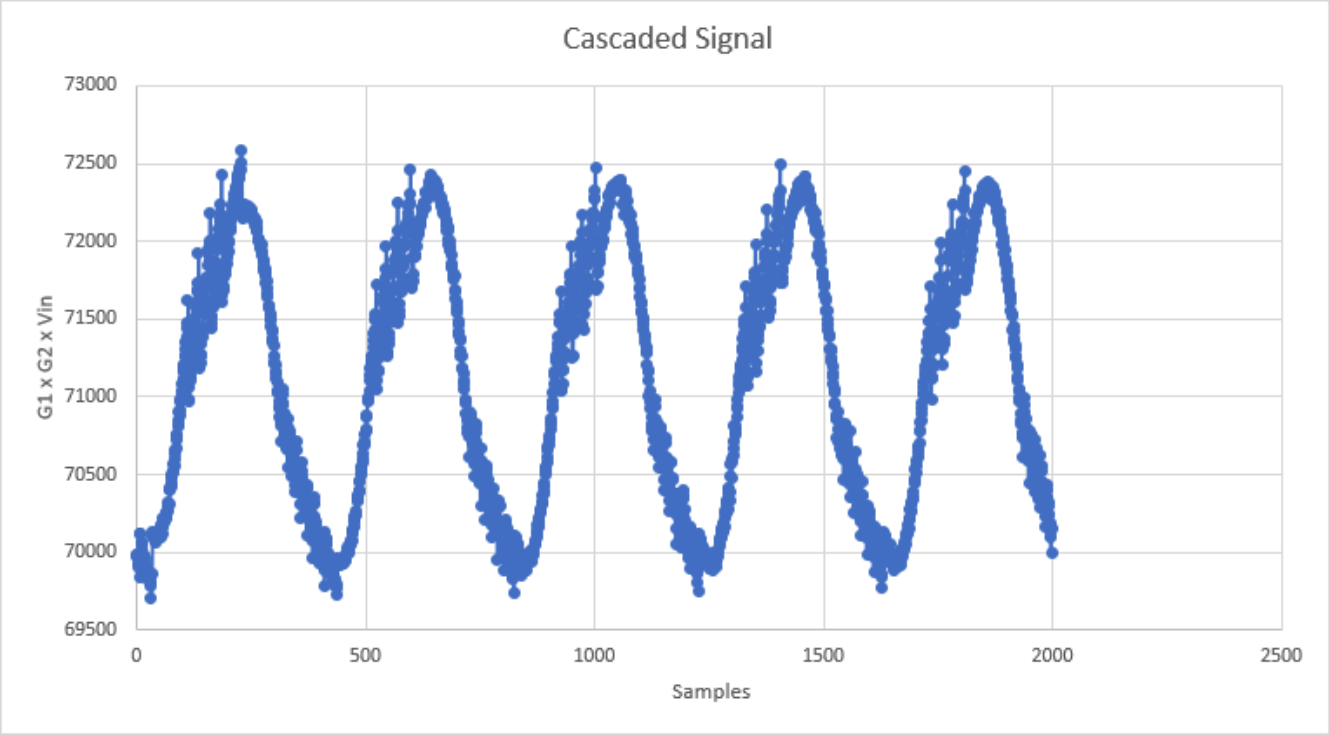

Figure 8-2. Cascade Signal

9 Additional Resources

- Texas Instruments, *Download the MSPM0 SDK*
- Texas Instruments, *Learn more about SysConfig*
- Texas Instruments, *MSPM0G LaunchPad™*
- Texas Instruments, *MSPM0 Academy*

Trademarks

All trademarks are the property of their respective owners.

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