Subsystem Design
Transimpedance Amplifier

TEXAS INSTRUMENTS

Design description

This subsystem demonstrates how to setup MSPM0 internal op-amps to a transimpedance amplifier (TIA) configuration and read the output with the internal ADC. The transimpedance op amp circuit configuration converts an input current source into an output voltage. The current to voltage gain is based on the feedback resistance. Download the code for this example.
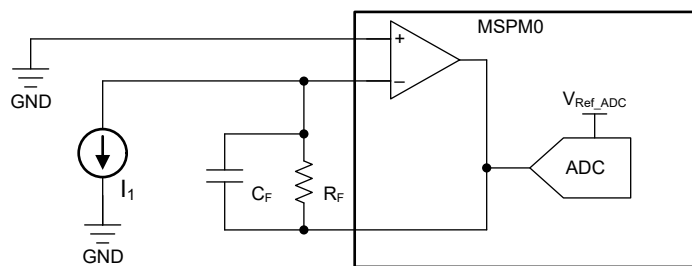
Figure 1-1 shows a functional diagram of this subsystem.

Figure 1-1. Subsystem Functional Block Diagram

Required peripherals

This application requires an integrated OPA and ADC.

Table 1-1. Required Peripherals

Sub-block Functionality	Peripheral Use	Notes
TIA (current to voltage translation)	(1x) OPA	Called "TIA_INST" in code
Analog signal capture	(1x) ADC12	Called "ADC12_0_INST" in code

Design steps

1. Calculate the gain resister, R_F

$$R_F = \frac{V_{Ref\_ADC} - V_{Min}}{I_{1Max}}$$

(1)

where

- V_{Ref_ADC} is the selected reference for the ADC peripheral
- V_{Min} is the minimum op amp output voltage
- I_{1Max} is the max current of the input current source

2. Calculate the feedback capacitor to meet the circuit bandwidth.

$$C_F \leq \frac{1}{2 \times \pi \times R_F \times f_p}$$

(2)

where f_p is the maximum frequency of the input current source.

3. Calculate the necessary op amp gain bandwidth (GBW) for the circuit to be stable.

$$GBW > \frac{C_i + C_F}{2 \times \pi \times R_F \times C_F^2}$$

(3)

where...

$$C_i = C_s + C_d + C_{cm}$$

(4)

given:

- C_S : Input source capacitance
- C_d : Differential input capacitance of the amplifier. This can generally be estimated at 3 pF for MSPM0 devices.
- C_{cm} : Common-mode input capacitance of the inverting input

4. Determine OPA GBW settings that can be utilized by comparing lower limit to calculated value in step 3.
5. Setup OPA in SysConfig for external connections for circuit.
6. Setup ADC in SysConfig for internal connection to chosen OPA output.
7. Set ADC sample time in SysConfig to a minimum of t_{Sample_PGA} as given in the device data sheet.

Design considerations

1. OPA supply is the VCC of the MSPM0.
2. OPA GBW setting: A lower GBW setting for the OPA consumes less current but responds more slowly. conversely, a higher GBW consumes more current, but has a larger slew rate and faster enable and settling times. See the device-specific data sheet for specification differences between the modes.
3. OPA noninverting input: Instead of GND potential, the OPA noninverting input can be given a small bias voltage to keep the output from saturating to GND if a current source is not active (such as when a photodiode is in a no-light condition). This can be accomplished through external voltage input, or through internal peripherals such as the DAC12 or DAC8 inside the COMP peripheral. In the latter case, the pin associated with the OPA noninverting input can be used for other purposes.
4. ADC sampling: This example continuously samples the OPA output. If this is not desired, a timer can be used to set a fixed interval of sampling.
5. ADC results: This example only stores the most current result captured in the global variable *gADCResult* . Full applications can store several readings in an array before performing actions on the data.
6. ADC reference selection: MSPM0 devices can provide a reference voltage to the ADC from the internal reference generator (VREF), external source, or MCU VCC. See the device-specific data sheet for options available for your chosen device. The reference voltage chosen sets the full scale range the ADC can sample and must accommodate the maximum OPA output voltage.
7. Race conditions on gCheckADC: This application clears gCheckADC as soon as possible. If the application waits too long to clear gCheckADC it can inadvertently miss new data.

Software flowchart

Figure 1-2 shows the code flow diagram for this example and explains how the ADC samples the OPA output.
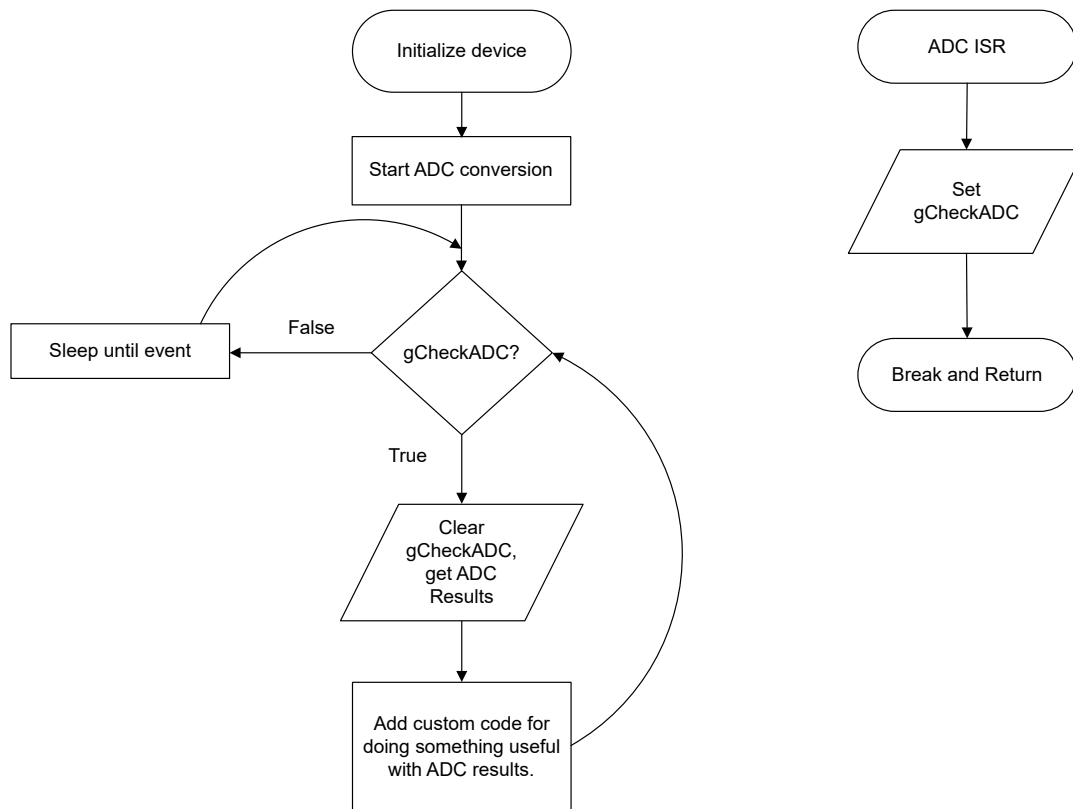

Figure 1-2. Application Software Flowchart

Device configuration

This application makes use of TI System Configuration Tool (SysConfig) graphical interface to generate the configuration code for the OPA and ADC. Using a graphical interface to configure the device peripherals streamlines the application prototyping process.

Application code

The code for what is described in Figure 1-2 can be found in the beginning of *main()* in the *TIA_Example.c* file. The following code snippet shows where to add custom code to perform useful actions after obtaining the ADC results of the measured current source. It is up to the user to determine what actions to take and to correlate ADC results with current source activity. For example, if connected to a photodiode, a design might average the ADC results to ignore small fluctuations of light and perform a delta calculation to detect large changes of light.

```
while (1) {
    DL_ADC12_startConversion(ADC12_0_INST);
    while (false == gCheckADC) {
            __WFE();
    }
    /* * This is where the ADC result is grabbed from ADC memory.
    * A user may want to modify this to place multiple results into an array,
    * or add code to perform additional calculations or filters to data obtained.
    */
    gADCResult = DL_ADC12_getMemResult(ADC12_0_INST, DL_ADC12_MEM_IDX_0);
    gCheckADC = false;
    DL_ADC12_enableConversions(ADC12_0_INST);
}
```

Additional Resources
- Download the MSPM0 SDK
- Learn more about SysConfig
- MSPM0L LaunchPad development kit
- MSPM0G LaunchPad development kit
- MSPM0 Timer academy
- MSPM0 ADC academy
- MSPM0 OPA academy

Revision History

Trademarks

All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER