AN-1543
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

SensorStrobe and Input Sampling for Time Synchronized, Sensor Data Sampling in the ADuCM4050

INTRODUCTION

This application node describes SensorStrobe™ and input sampling, a mechanism from Analog Devices, Inc., that recognizes the low power, consistent, and synchronized data acquisition from sensors. The SensorStrobe and input sampling technology are available on the ADuCM4050, which allows the synchronized data sampling of sensors that are connected to the device.

Precise sampling of a sensor that synchronizes to an accurate time base is a requirement in variety of sensor node applications. The sampling of the sensor data is dictated by the microcontroller unit (MCU). In the traditional approach, the software on the MCU generates pulses on a general-purpose input/output (GPIO), which triggers the sensor at specific intervals to collect data. This approach involves a considerable amount of software overhead, which increases the current consumption of the device because the device is active most of the time. The triggering pulse can drift over time due to software dependency.

The SensorStrobe and input sampling capabilities of the ADuCM4050 address the issues of the traditional software approach with the following features. SensorStrobe and input sampling work in hibernate mode, the lowest power mode in the ADuCM4050, resulting in low current consumption that extends battery life. SensorStrobe and input sampling do not intervene with the software, and pulse triggering is independent of other software executions, which gives an accurate pulse generation.

This application note uses an example setup that consists of the ADuCM4050 connected to Reed switches. The SensorStrobe mechanism powers the Reed switches, and input sampling enables data collection when the entire system is in the lowest power mode. The reduction in power consumption is evident in comparison to the traditional approach that does not utilize SensorStrobe and input sampling.

TABLE OF CONTENTS

REVISION HISTORY

6/2018—Revision 0: Initial Version

SensorStrobe

The SensorStrobe technology samples sensors in an efficient, low power, intrinsically synchronized manner, without MCU involvement. The ADuCM4050 supports SensorStrobe, and can be used in active, flexi, and hibernate modes. SensorStrobe allows the ADuCM4050 to be in hibernate mode when the sensor or device collects data at periodic intervals.

SensorStrobe is an alarm function on the real-time clock (RTC1) on the ADuCM4050 that sends an output pulse to an external device or sensor via a GPIO pin. The SensorStrobe channels on the ADuCM4050 can be connected to one or multiple sensors that can be activated or enabled to perform sensing or conversion. The SensorStrobe events are scheduled by the CPU by instructing RTC1 to activate the SensorStrobe events at a specific target time relative to the RTC real-time count. When RTC1 is enabled, the CPU can go into hibernate mode and wait for event occurrence.

The following are the paramount attributes of the SensorStrobe feature on the ADuCM4050:

- Three independent SensorStrobe channels. All channels are 16-bit, and the slowest output pulse is 0.5 Hz (for a 32.768 kHz RTC clock). The granularity of the output pulse is one RTC (30.51 µs for a 32.768 kHz RTC clock). The high and low durations (16 bits each) of output pulses are configurable for every channel.
- Each of the three SensorStrobe channels has a readable, sticky interrupt source bit that activates (sticks high) when a SensorStrobe event occurs. A SensorStrobe event can be rising edge, falling edge, or both edges of an output channel. This interrupt source bit can be enabled to interrupt the processor and can also wake the processor up from low power mode.
- The SensorStrobe channels can be reconfigured and enabled or disabled during operation, with no interruption to channels that are not concerned with the reconfiguration.
- The SensorStrobe channel output pulse polarity can be changed during operation with no interruption to the other SensorStrobe channels.
- All SensorStrobe channels support fine control over periods and duty cycles by enabling or disabling the autoreloading from the high duration register (RTC1_SSxHIGHDUR) and low duration register (RTC1_SSxLOWDUR) when a SensorStrobe event occurs.
- Three SensorStrobe channels (RTC1_SSx) can send out three different signals on three different GPIO pins. A pulse on one SensorStrobe channel has no effect on any of the other SensorStrobe channels.
- Each SensorStrobe channel has a dedicated output GPIO that an output pulse can be exported through. The exported signal can be inverted on the same GPIO.

Table 1 describes the SensorStrobe GPIO pin location on the ADuCM4050.

Table 1. GPIO Pin Location for SensorStrobe

Channel Name	GPIO Pin No.	Pin Location on EV-GEAR-EXPANDER1Z[1]
RTC1_SS1	P2_11	P8-6
RTC1_SS2	P1_12	A5-10
RTC1_SS3	P2_08	A5-8

[1] Connect the EV-GEAR-EXPANDER1Z to the EV-COG-AD4050LZ to access all of the GPIO pins.

USING SensorStrobe WITH THE ADXL362

A system example is shown in Figure 1 to demonstrate the benefits of SensorStrobe. This system consists of the EV-COG-AD4050LZ development board with the ADuCM4050 and the ADXL362 on-board.

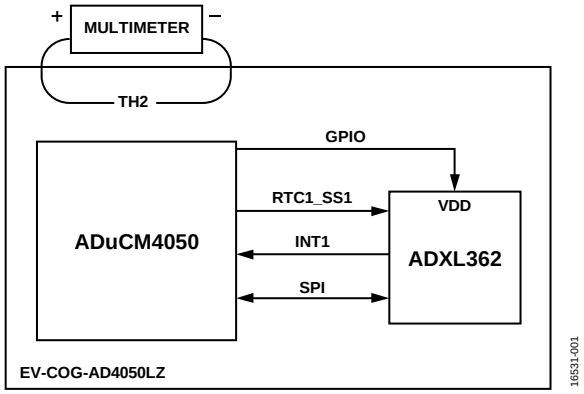

Figure 1. System Connection on the EV-COG-AD4050LZ Board for Current Measurement

The SensorStrobe on the ADuCM4050 is combined with the external triggering feature of the ADXL362 to collect accelerometer data at the lowest possible system power consumption. SensorStrobe allows the ADuCM4050 to be in hibernate mode when the ADXL362 collects data at periodic intervals.

The ADXL362 is an ultra low power, 3-axis, microelectromechanical system (MEMS) accelerometer. The device supports a 512-sample, first in, first out (FIFO) buffer to store sensor data. The large FIFO buffer saves power at the system level by extending the hibernate period and simultaneously storing large accelerometer data. The ADXL362 also supports external triggering mode at the INT2 pin.

The ADuCM4050 generates trigger pulses on the RTC1_SS1 channel, which is connected to the INT2 pin of the ADXL362 on the EV-COG-AD4050LZ development board.

The ADXL362 is configured for measurement mode to interrupt the MCU when the FIFO watermark is reached. The ADuCM4050 configures ADXL362 in external triggering mode and SensorStrobe to generate triggering pulse at 128 Hz. After the configuration is complete, ADuCM4050 is placed in hibernate mode and waits for a FIFO watermark interrupt from the ADXL362.

On every pulse, the ADXL362 samples the accelerometer data and stores the data in the FIFO buffer. When the FIFO watermark is reached, the ADXL362 interrupts the ADuCM4050 via the SYS_WAKE1 (P1_00) pin. The ADuCM4050 exits hibernate mode, processes this interrupt, and uses the read mode feature of the ADXL362 to drain the entire FIFO in a single command, minimizing the serial peripheral interface (SPI) protocol overhead. The direct memory access (DMA) controller in the ADuCM4050 can drain the ADXL362 FIFO buffer. The ADuCM4050 enters flexi mode when the DMA is active and performs the data transfer, further reducing the active period and system power consumption.

SensorStrobe SOFTWARE FLOWS

This section describes the software flows in the SensorStrobe example system, shown in Figure 2 and Figure 3.

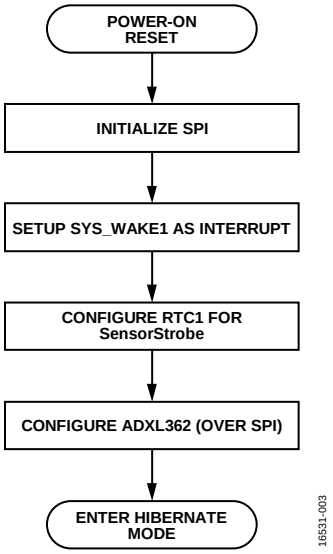

Figure 2. Initialization and Configuration Steps

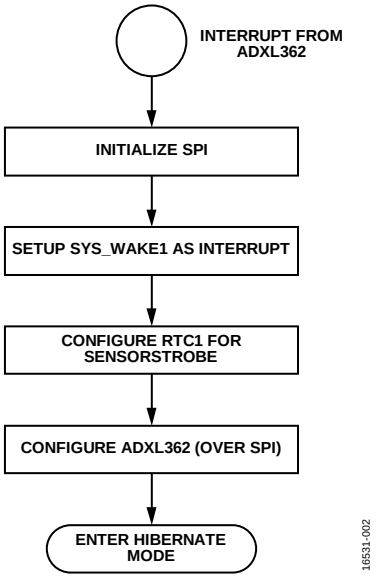

Figure 3. Data Collection from the ADXL362

INPUT SAMPLING

Input sampling is a monitoring feature and associated with SensorStrobe on the ADuCM4050, where the MCU is only involved/active when the state of the device or sensor is critical or interesting and needs processing. Otherwise, the system stays in low power mode.

When input sampling is enabled, the input data from the external device or sensor is sampled at a desired SensorStrobe sampling frequency at configurable sampling point. When a specific sequence from an external device satisfies one of the configurable sampling pattern matching conditions, the RTC on the ADuCM4050 can optionally send an interrupt to wake up the MCU or to process the sensor data. The processing of sensor data sampling and matching with nominated sampling pattern conditions continues while MCU remains in low power mode. This results in a reduction of the overall active period and power consumption of the system.

The following are the paramount attributes of the input sampling feature on the ADuCM4050:

- Input sampling can collect/sense data from an external device/sensor periodically and the SensorStrobe output pulse is the sampling frequency.
- Input sampling has a total of 8 GPIO input pins available, out of which a maximum of three (GPx_IN0, GPx_IN1, and GPx_IN3) can be configured/selected by each of the input sampling channels.
  Always drive all of the configured/selected GPIO pins to some proper state, irrespective of output pulse, for example, RTC1_SSx being high or low. Never leave the GPIO configured as an input to the ADuCM4050 floating. Table 2 describes the input sampling GPIO pin locations.

Table 2. Input Sampling GPIO Pin Location

SensorStrobe Channel	Input Sampling GPIO Pins	Pin Location on EV-GEAR-EXPANDER1Z[1]
RTC1_SS1	P0_00 (GP1_IN0)	A5-5
	P0_12 (GP1_IN1)	P8-7
	P1_02 (GP1_IN2)	P6-2
RTC1_SS2	P0_08 (GP2_IN0)	A4-1
	P1_13 (GP2_IN1)	P6-9
	P0_09 (GP2_IN2)	A4-4[2]
RTC1_SS3	P2_07 (GP3_IN0)	A2-5
	P2_09 (GP3_IN1)	P6-10

[1] Connect the EV-GEAR-EXPANDER1Z to the EV-COG-AD4050LZ to access all of the GPIO pins.
[2] Jumper change is required on the EV-COG-AD4050LZ baseboard. Short the JP2 jumper, Pin 2 and the JP2 jumper, Pin 4.

- Input sampling supports configurable sampling points and inputs can be sampled in the following instances:
  - Before rising edge sampling (BRES)
  - After rising edge sampling (ARES)
  - Before falling edge sampling (BFES)
  - After falling edge sampling (AFES)
  - Both edges of SensorStrobe; falling edge sampling (FES) and/or rising edge sampling (RES).

  Note that if the ARES point is selected, all GPIO pins configured with that SensorStrobe channel are sampled for one RTC clock (for example, 30.5 µS) after the rising edge of the SensorStrobe pulse. A write to the RTC1_CR6SSS register can enable the desired sampling point for the corresponding enabled input sampling channel.
- The sampled value is available as a readback in the RTC1_SR7.SSxSMP register (x denotes the input sampling channel) for each of the enabled channels. This register updates when the next sampling point is reached. Read sampled data within the next sampling point to avoid overlooking data.
- Input sampling can optionally send an interrupt to the ADuCM4050 based on the sampling pattern matching conditions for each RTC1_CR5SSS.SSxSMPEN enabled channel, independently. This interrupt reduces the involvement of the MCU to make a decision, therefore, the MCU is only used to process the sensor data when an interesting scenario occurs. The following are the sampling pattern matching conditions that can receive an interrupt from input sampling:
  - Matching Condition 1. Current sample ($SMPx_t$) is the same as the previously sampled value ($SMPx_{t-1}$).
  - Matching Condition 2. Current sample is different from the previously sampled value.
  - Matching Condition 3.Current sample matches an expected sample value (EXP_SMPx).
  - Matching Condition 4. Current sample does not match an expected sample value.
- Each input sampling channel has a separate expected sample value register (RTC1_CR7SSS.SSxSMPEXP), which is used to store a user defined value that is expected from the external sensor, and is of interest from an application perspective. Every new sample is compared with the expected sample value, and depending upon the sampling pattern matching condition as configured in the RTC1_CR7SSS.SSxSMPPTRN register, an interrupt is generated to trigger the ADuCM4050.
- In the event of GPIO sampling activity matching with the sampling pattern, the sticky interrupt status bit is set, which can be cleared by writing a 1 to the bit.

- If input sampling is enabled for one SensorStrobe channel, the sampling and channel are independent to other SensorStrobe channels. This means that one SensorStrobe channel can be configured as input sampling and the other channels can only be used for SensorStrobe, or be disabled.

INPUT SAMPLING APPLICATION DEMO

The data sampling demonstration application using the Reed switches showcases the use of SensorStrobe and input sampling features together on the ADuCM4050 using the EV-COG-AD4050LZ development board.

In Figure 4, the Reed Switches are powered using the SensorStrobe pin (RTC1_SS1), and the input sampling GPIOs (GP1_IN1 and GP1_IN2, see Table 2) are used to sample the incoming data. A bar magnet is mounted on one end of a propeller shaft. This propeller shaft assembly is inserted inside of a water pipe. The magnets are placed in close vicinity to the Reed switches, and the Reed switches are placed 45° angle to each other to produce a phase difference that determines the direction of the water flow.

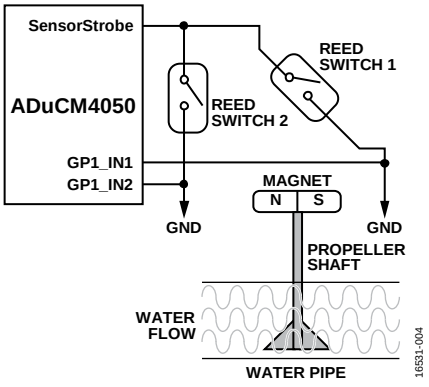

Figure 4. Low Power Water Metering System Overview Using the ADuCM4050

During a power-on reset (POR) of the system, the ADuCM4050 configures SensorStrobe Channel 1, enables input sampling and configures the system for input sampling mode. After this configuration, the triggering pulse is sent from the RTC1_SS1 pin, which powers-on the Reed switch and input sampling simultaneously samples the data. The configurable sampling point in this demo is set to ARES and the sampling pattern match condition is set to Matching Condition 2 (current sample is different from the previously sampled value).

The water flow inside the pipe rotates the propeller, which rotates the magnet. The Reed switches close when the north-south (NS) pole of the magnet is aligned parallel to any one of the Reed switches. This generates a Logic 1 on the Reed switch output, and this logic is available at the corresponding input sampling GPIO.

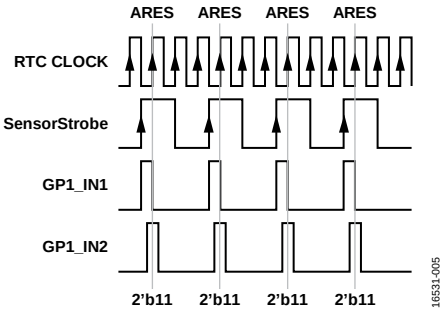

Figure 5. Input Sampling for Two GPIO Inputs from Reed Switch 1 and Reed Switch 2

For a first time calculation of the water flow, the sampling pattern match condition is set to Matching Condition 3 (current sample matches an expected sample value), and the expected sample value is set to 2'b11. When there is a match between the current sample value and the expected sample value, for example, $SMPx_t$ = EXP_SMPx = 2'b11, the MCU wakes up, increments the count value of 2'b11, and then enters hibernate mode. This process continues for 1 sec and the MCU counts the number of 2'b11 pulses on the GP1_IN1 and GP1_IN2 pins (see Table 2). The number of 2'b11 pulses on these pins is used to calculate the frequency of the incoming pulses of the Reed switches, which indirectly relates to the rotation of the magnet and propeller, as shown in Equation 1. The SensorStrobe frequency is changed to integer multiple of the recently calculated magnet rotation frequency.

$$f = 1 \div number\ of\ 2'b11\ pulses \quad (1)$$

When the initial magnet rotation frequency is calculated, the sampling pattern match condition is changed to Matching Condition 2 (current sample is different from the previously sampled value). The change in the SensorStrobe frequency, for example, the sampling frequency, ensures that the data collected by the input sampling at ARES of the SensorStrobe pulse is the same as the previously sampled data. Per the sampling pattern matching condition, the MCU only wakes up if the current sample is different from the previous sample. If this condition is satisfied, the MCU wakes up, calculates the new magnet rotation frequency at the GP1_IN1 and GP1_IN2 pins (see Table 2), and enters hibernate mode after configuring the SensorStrobe frequency to the newly calculated value. This also means that the water flow rate in the pipe has changed, therefore, the system must calculate the new water flow rate to measure the volume of water consumed for billing process. Because the MCU only wakes up or is involved when there is change in the water flow rate, the involvement of the MCU and the power consumption is reduced.

INPUT SAMPLING SOFTWARE FLOW

Figure 6 illustrates the software flow in the input sampling system example.

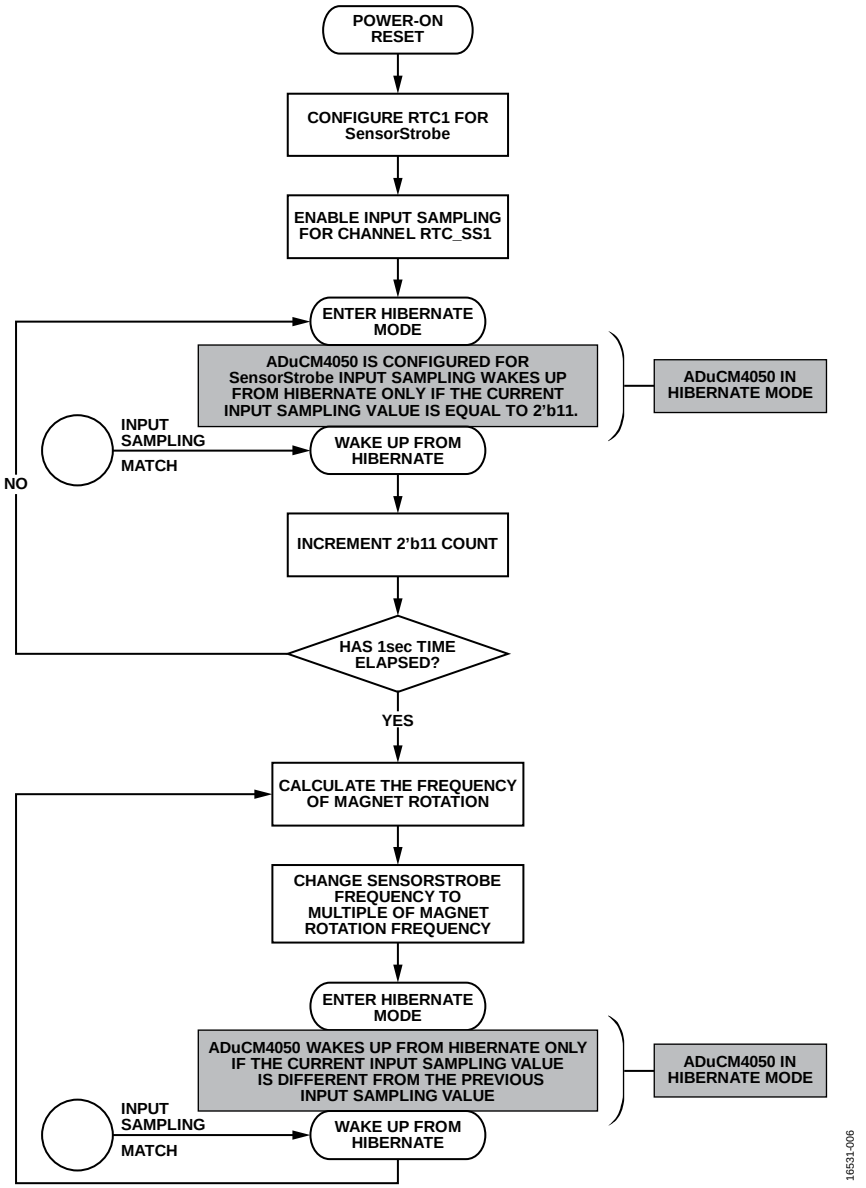

Figure 6. Input Sampling Flow Chart for a Low Power Water Metering System Example

SOURCE CODE FUNCTION

Figure 7 and Figure 8 show the source code that is used to configure the RTC1 for SensorStrobe and input sampling.

```
////////////////////////////////////////////////////////////////////////////////
//////////////////////// Configure RTC1 SensorStrobe Channels //////////////////////
////////////////////////////////////////////////////////////////////////////////
/*
    @detail             RTC1_SS_Config will Enable/Disable each of the RTC1 SS
                        Channel and set the HIGH and LOW Duration value.

    @param[in]    ch    Select RTC SS Channels
                        0 - Disable All Channels
                        1 - Select Channel 1
                        2 - Select Channel 2
                        3 - Select Channel 3

    @param[in]    H     HIGH Duartion for the selected channel in "ch".

    @param[in]    L     LOW Duartion for the selected channel in "ch".
    @param[in]    POL   Option to Invert SS signal for the channel in "ch"
                        0 - No Inversion
                        1 - Invert Output
*/
void RTC1_SS_Config (int ch, uint32_t  H, uint32_t  L, int POL)
```

16531-007

Figure 7. Source Code to Configure RTC1 SensorStrobe

```
////////////////////////////////////////////////////////////////////////////////
//////////////////////  Configure RTC1 Input Sampling   ////////////////////////
////////////////////////////////////////////////////////////////////////////////
/*
    @detail             RTC1_IPS_Ch will Enable/Disable each of the RTC Input
                        Sampling for each of the RTC1 SS Channels. It will enable
                        Sample Activity Interrupt which sets HIGH when Expeted
                        Value in EXP_VAL matched with the Current Value.

    @param[in]    ch    Select RTC1 SS Channels
                        0 - Disable All Channels
                        1 - Select Channel 1
                        2 - Select Channel 2
                        3 - Select Channel 3

    @param[in]    ip    Select number of Input Sampling for selected channel in "ch".
                        0 - No Input Selected
                        1 - Input 0 Selected
                        2 - Input 1 Selected
                        3 - Input 1 & 0 Selected
                        4 - Input 2 Selected
                        5 - Input 2 & 0 Selected
                        6 - Input 2 & 1 Selected
                        7 - Input 2, 1 & 0 Selected

    @param[in] EXP_VAL  Expected Value of Input Sampling for selected channel in "ch".
*/

void RTC1_IPS_Ch(uint32_t ch, uint32_t ip, uint32_t EXP_VAL)
```

16531-008

Figure 8. Source Code to Configure RTC1 Input Sampling

POWER MEASUREMENT

The following section describes how to monitor the current consumption of the system. Refer Figure 1, which shows the system connection to digital multimeter (DMM) for power measurement.

SENSORSTROBE MONITORING

1. Load the SensorStrobe application on the EV-COG-AD4050LZ development board.
2. Remove the shunt jumper on TH2.
3. Connect the positive terminal of the DMM to the left-hand terminal of TH2, and connect the negative terminal of DMM to the right-hand terminal of TH2.
4. Press the RESET button on the development board.
5. Monitor the current consumption on the DMM.

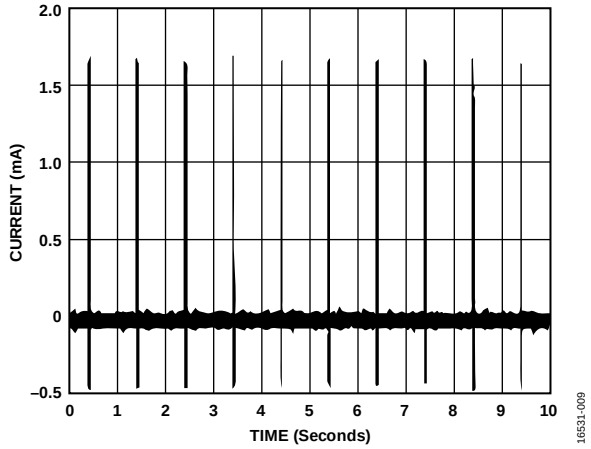

Figure 9. SensorStrobe Current Profile Working with the ADXL362

INPUT SAMPLING MONITORING

1. Load the low power water metering application on the EV-COG-AD4050LZ.
2. Remove the shunt jumper on TH2.
3. Connect the positive terminal of the DMM to left-hand terminal of TH2 and connect the negative terminal of the DMM to the right-hand terminal of TH2.
4. Press the RESET button on the development board.
5. Monitor the current consumption on the DMM.

During the current measurement for low power water metering application, the flow of water in the pipe changes every 5 sec so that the ADuCM4050 wakes up and performs a flow rate measurement (see Figure 10).

Figure 10. Input Sampling Current Profile Working with the Low Power Water Metering Application

AN16531-0-6/18(0)