

## ADuCM4050 での SensorStrobe と入力サンプリングによる時刻同期、 センサー・データ・サンプリング

### はじめに

このアプリケーション・ノートでは、低消費電力のセンサーによる一貫した同期データ・アキュジションを認識するメカニズムである SensorStrobe™ および入力サンプリングの機構について説明します。SensorStrobe および入力サンプリングの技法は、ADuCM4050 で使用でき、デバイスに接続されたセンサーによる同期データ・サンプリングが可能になります。

正確なタイム・ベースに同期させたセンサーによる高精度なサンプリングは、様々なセンサー・ノード・アプリケーションで必要とされます。センサー・データのサンプリングを指示するのは、マイクロコントローラ・ユニット (MCU) です。従来の方法では、MCU に搭載されたソフトウェアが汎用入出力 (GPIO) にパルスを生成し、特定の間隔でセンサーをトリガすることでデータを収集します。この方法では、ソフトウェアのオーバーヘッドが大きく、デバイスがほとんどの時間アクティブになるため、電流の消費量が増えます。パルスのトリガはソフトウェアに依存するため、時間と共にドリフトすることもあります。

ADuCM4050 の SensorStrobe および入力サンプリング技術は、以下の機能によって従来のソフトウェア方式の問題に対処します。SensorStrobe および入力サンプリングは、休止モードで動作します。これは、ADuCM4050 で最も低消費な電力モードであるため、消費電流を抑え、バッテリー寿命を延ばすことができます。SensorStrobe および入力サンプリングにソフトウェアが介在することはなく、パルスのトリガは実行中の他のソフトウェアから独立しているため、正確なパルス生成が可能です。

このアプリケーション・ノートでは、リード・スイッチを ADuCM4050 に接続した構成をセットアップ例として使用します。システム全体が最も低消費な電力モードの状態で、SensorStrobe 機構がリード・スイッチを駆動し、入力サンプリングがデータ収集を実行します。SensorStrobe および入力サンプリングを使用しない従来の方法と比較して、明らかに消費電力を削減できます。

## 目次

はじめに .....	1	入力サンプリングのアプリケーション実証例 .....	6
改訂履歴 .....	2	入力サンプリングのソフトウェア・フロー .....	7
SensorStrobe .....	3	ソース・コードの構成 .....	8
ADXL362 に接続して使用する SensorStrobe .....	3	電力測定 .....	9
SensorStrobe のソフトウェア・フロー .....	4	SensorStrobe のモニタリング .....	9
入力サンプリング .....	5	入力サンプリングのモニタリング .....	9

## 改訂履歴

6/2018—Revision 0: Initial Version

## SensorStrobe

SensorStrobe 技術は、MCU が介在することなく、高効率かつ低消費電力で、本質的に同期された方法でセンサーによるサンプリングを実行します。ADuCM4050 は SensorStrobe をサポートし、アクティブ・モード、flexi モード、休止モードで使用できます。SensorStrobe を使用すると、センサーやデバイスが一定の間隔でデータを収集中に、ADuCM4050 を休止モードに移行できます。

SensorStrobe は、ADuCM4050 に搭載されたリアルタイム・クロック (RTC1) のアラーム機能で、GPIO ピンを通じて外部のデバイスやセンサーに出力パルスを送信します。ADuCM4050 の SensorStrobe チャンネルを 1 つまたは複数のセンサーに接続して、センシングや変換機能をアクティブ化または有効化できます。SensorStrobe のイベントは、RTC のリアルタイム・カウンタを基準とする特定の目標時間で SensorStrobe のイベントをアクティブにするよう CPU から RTC1 に命令することにより、スケジュールされます。RTC1 がイネーブルになると、CPU は休止モードに入ってイベントの発生に備え待機することができます。

ADuCM4050 に搭載された SensorStrobe 機能の重要な特性を下記に示します。

- 3つの独立した SensorStrobe チャンネル。すべてのチャンネルは 16 ビットで、最も遅い出力パルスは 0.5Hz (32.768kHz の RTC クロック使用時) です。出力パルスの粒度は 1RTC (32.768kHz の RTC クロックで 30.51 $\mu$ s) です。出力パルスのハイとローの時間 (それぞれ 16 ビット) は、すべてのチャンネルで設定可能です。
- 3つの SensorStrobe チャンネルそれぞれに、読み出し可能な割り込み源のステイキ・ビットがあり、SensorStrobe のイベントが発生するとアクティブ (ハイに固定) になります。SensorStrobe のイベントは、出力チャンネルの立上がりエッジ、立下がりエッジ、または両エッジで発生します。この割り込み源ビットは、プロセッサへの割り込みをイネーブルできると共に、プロセッサを低消費電力モードからウェークアップさせることもできます。
- SensorStrobe チャンネルは、動作中に再設定やイネーブル/ディスエーブルが可能なため、再設定に関係のないチャンネルへの割り込みをする必要がありません。
- SensorStrobe チャンネルの出力パルス極性は、他の SensorStrobe チャンネルへの割り込みをすることなく動作中に変更できます。
- すべての SensorStrobe チャンネルは、SensorStrobe イベントが発生したとき、全周期及び全デューティ・サイクルをハイ時間レジスタ (RTC1\_SSxHIGHDUR) およびロー時間レジスタ (RTC1\_SSxLOWDUR) からの自動再ローディングを有効化/無効化することで、微調整することが可能です。
- 3つの SensorStrobe チャンネル (RTC1\_SSx) は、3本の異なる GPIO ピンから3つの異なる信号を送信することができます。1つの SensorStrobe チャンネルのパルスは、他の SensorStrobe チャンネルに影響を与えません。

- 各 SensorStrobe チャンネルは、専用の出力 GPIO を備えています。出力されたパルス信号は、反転させることもできます。

表 1 に ADuCM4050 の SensorStrobe 用の GPIO ピン位置を示します。

表 1. SensorStrobe 用の GPIO ピン位置

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

<sup>1</sup> EV-GEAR-EXPANDER1Z を EV-COG-AD4050LZ に接続することで、すべての GPIO ピンにアクセスできます。

### ADXL362 に接続して使用する SensorStrobe

SensorStrobe のメリットを実証するシステム例を図 1 に示します。このシステムは、ADuCM4050 と ADXL362 を搭載した EV-COG-AD4050LZ 開発ボードで構成されています。

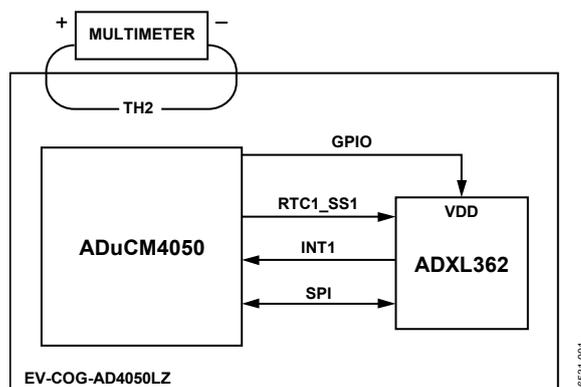


図 1. 電流測定用の EV-COG-AD4050LZ ボードのシステム接続

ADuCM4050 の SensorStrobe と ADXL362 の外部トリガ機能を組み合わせることで、システムの消費電力をできる限り抑えて加速度センサーのデータを収集します。SensorStrobe を使用することで、ADXL362 が一定間隔でデータを収集しているときに ADuCM4050 を休止モードに移行できます。

ADXL362 は超低消費電力の 3 軸マイクロマシン (MEMS) 加速度センサーです。このデバイスには、センサー・データを保存するために、512 サンプルの先入れ先出し (FIFO) バッファが内蔵されています。この大容量 FIFO バッファにより、CPU の休止モードの時間を延長して多くの加速度センサー・データを保存することができるため、システム・レベルで消費電力が削減されます。また、ADXL362 は INT2 ピンで外部トリガ・モードにも対応します。

ADuCM4050 は、RTC1\_SS1 チャンネルにトリガ・パルスを生成します。このチャンネルは EV-COG-AD4050LZ 開発ボード上で ADXL362 の INT2 ピンに接続されています。

ADuCM4050 は ADXL362 を外部トリガ・モードに構成し、SensorStrobe を 128Hz のトリガ・パルスが生成できるように構成します。構成が完了すると、ADuCM4050 は休止モードに入り、ADXL362 から FIFO ウォーターマーク割込みが実行されるまで待機します。

SensorStrobe の各トリガ・パルス発生時に、ADXL362 は加速度センサー・データをサンプリングして FIFO バッファにデータを保存します。FIFO サンプル数がウォーターマークに到達すると、ADXL362 は SYS\_WAKE1 (P1\_00) ピン経由で ADuCM4050 に割込みを実行します。ADuCM4050 は休止モードから復帰して割込みを処理します。ADXL362 の読出しモード機能を使用して 1 回のコマンドで FIFO 全体をドレインすることで、シリアル・ペリフェラル・インターフェース (SPI) プロトコルのオーバーヘッドを最小限に抑えます。ADuCM4050 に搭載されたダイレクト・メモリ・アクセス (DMA) コントローラにより、ADXL362 の FIFO バッファをドレインできます。DMA がアクティブになってデータ転送を実行すると、ADuCM4050 は Flexi モードに入り、アクティブな時間とシステムの消費電力を更に低減します。

## SensorStrobe のソフトウェア・フロー

ここでは、図 2 と図 3 に示すように、SensorStrobe のサンプル・システムにおけるソフトウェア・フローを説明します。

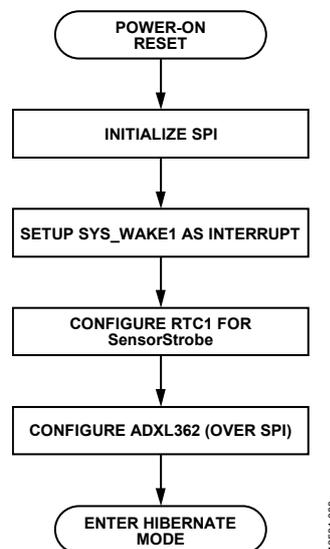


図 2. 初期化および構成の手順

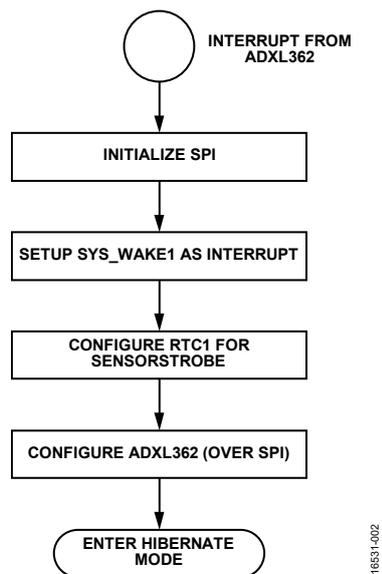


図 3. ADXL362 からのデータ収集

## 入力サンプリング

入力サンプリングは、ADuCM4050 の SensorStrobe を対象としたモニタリング機能で、MCU は、デバイスまたはセンサーが危険あるいは重要な状態のために処理が必要となる場合にのみ、関与またはアクティブ化します。それ以外の場合、システムは低消費電力モードを維持します。

入力サンプリングを有効にすると、必要な SensorStrobe サンプリング周波数、および任意に設定されたサンプリング・ポイントで、外部デバイスまたはセンサーからの入力データをサンプリングします。設定可能なサンプリング・パターン条件の 1 つに一致する特定のシーケンスが外部デバイスから入力された場合、ADuCM4050 の RTC は、MCU をウェークアップさせるかセンサー・データを処理するために、割込みを送信することもできます。センサー・データのサンプリング処理、および指定したサンプリング・パターン条件とのマッチングは、MCU が低消費電力モードのまま実行し続けます。これにより、システムの全アクティブ時間と消費電力を低減できます。

ADuCM4050 に搭載された入力サンプリング機能の重要な特性を下記に示します。

- 入力サンプリングは、外部デバイス/センサーから定期的にデータの収集・検出を行います。SensorStrobe の出力パルスがサンプリング周波数です。
- 入力サンプリングには、全部で 8 本の GPIO 入力ピンを使用可能で、その中から各入力サンプリング・チャンネルにつき最大 3 本 (GPx\_IN0、GPx\_IN1、GPx\_IN3) を選択/設定できます。  
選択/設定したすべての GPIO ピンは常に、出力パルスに関係なく適切な状態 (例えば RTC1\_SSx をハイまたはロー) にしなければなりません。ADuCM4050 への入力として設定した GPIO は、決してフローティング状態にしないでください。表 2 に、入力サンプリング用 GPIO ピンの位置を示します。

表 2. 入力サンプリング用 GPIO ピンの位置

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

<sup>1</sup> EV-GEAR-EXPANDER1Z を EV-COG-AD4050LZ に接続することで、すべての GPIO ピンにアクセスできます。

<sup>2</sup> EV-COG-AD4050LZ ベースボードのジャンパを変更する必要があります。JP2 ジャンパのピン 2 と JP2 ジャンパのピン 4 を短絡します。

- 入力サンプリングでは、サンプリング・ポイントを設定することが可能で、入力は以下の状態のときにサンプリングできます。
  - 立上がりエッジ前のサンプリング (BRES)
  - 立上がりエッジ後のサンプリング (ARES)
  - 立下がりエッジ前のサンプリング (BFES)
  - 立下がりエッジ後のサンプリング (AFES)
  - SensorStrobe の両エッジ、立下がりエッジでのサンプリング (FES) または立上がりエッジでのサンプリング (RES)。
- ARES ポイントを選択した場合、すべての GPIO ピンは、SensorStrobe パルスの立上がりエッジ後、1RTC クロック (例えば 30.5μs) で SensorStrobe チャンネルがサンプリングされるように設定されます。RTC1\_CR6SSS レジスタに書き込むことにより、イネーブルされた入力サンプリング・チャンネルにおいて所望のサンプリング・ポイントをイネーブルにできます。
- サンプリングされた値は、イネーブルされた各チャンネルの RTC1\_SR7.SSxSMP レジスタ (x は入力サンプリング・チャンネルを表します) のリードバックとして入手できます。このレジスタは、次のサンプリング・ポイントに達すると更新されます。データの読み落としがないように、サンプリングされたデータは次のサンプリング・ポイントまでに読み出してください。
- 入力サンプリングは、オプションとして、RTC1\_CR5SSS.SxSMPEN でイネーブルされたチャンネルごとの独立したサンプリング・パターンのマッチング条件に基づき、ADuCM4050 に割込みを送信することもできます。この割込みにより MCU が関与して判断を下す必要が減り、MCU がセンサー・データの処理に使用されるのは、重要な状況が発生した場合のみになります。以下に、入力サンプリングから割込みを受信する可能性のあるサンプリング・パターンのマッチング条件を示します。
  - マッチング条件 1. 現在のサンプル (SMPx<sub>i</sub>) がその前にサンプリングした値 (SMPx<sub>i-1</sub>) と等しい。
  - マッチング条件 2. 現在のサンプルがその前にサンプリングした値と異なる。
  - マッチング条件 3. 現在のサンプルが予想サンプル値 (EXP\_SMPx) と一致する。
  - マッチング条件 4. 現在のサンプルが予想サンプル値と一致しない。
- 各入力サンプリング・チャンネルには、個別の予想サンプル値レジスタ (RTC1\_CR7SSS.SSxSMPEXP) があり、外部センサーの予想サンプル値、およびアプリケーションの観点から参照したい値をユーザ定義の値として保存できます。新しいサンプルはすべて予想サンプル値と比較され、RTC1\_CR7SSS.SSxSMPPTRN レジスタで設定されたサンプリング・パターンのマッチング条件に応じて、ADuCM4050 をトリガする割込みが生成されます。
- GPIO サンプリングにサンプリング・パターンと一致するイベントが発生すると、割込みステータスのスティッキー・ビットがセットされます。このビットは、1 を書き込むとクリアできます。

- 1つの SensorStrobe チャンネルに対して入力サンプリングが有効化されると、このサンプリングとチャンネルは他の SensorStrobe チャンネルとは独立することになります。これにより、1つの SensorStrobe チャンネルを入力サンプリングとして設定し、他のチャンネルを SensorStrobe のみに使用したり、ディスエーブルにしたりできます。

## 入力サンプリングのアプリケーション実証例

リード・スイッチを使用したデータ・サンプリングの実証アプリケーションで、EV-COG-AD4050LZ 開発ボードを用いて ADuCM4050 に備わっている SensorStrobe および入力サンプリング機能の使用法を示します。

図 4 では、SensorStrobe ピン (RTC1\_SS1) を使用してリード・スイッチを駆動し、入力サンプリング用 GPIO (GP1\_IN1 および GP1\_IN2、表 2 参照) を使用して入力データのサンプリングを行っています。プロペラ・シャフトの一端には棒磁石が取り付けられています。このプロペラ・シャフトのアセンブリが、水道管の内部に挿入されています。磁石はリード・スイッチのすぐ近くに配置します。リード・スイッチは、互いに 45° の角度で配置して位相差を生成することで水流の方向がわかるようにします。

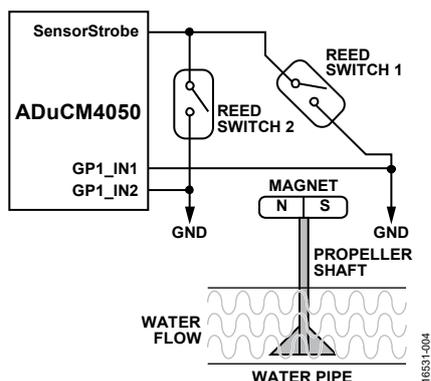


図 4. ADuCM4050 を使用した低消費電力水量計システムの概略図

システムのパワーオン・リセット (POR) の間に、ADuCM4050 は SensorStrobe Channel 1 を構成すると共に入力サンプリングを有効化し、システムを入力サンプリング・モードに構成します。構成が完了すると、RTC1\_SS1 ピンからトリガ・パルスを送信してリード・スイッチを起動すると同時に、入力サンプリングによってデータのサンプリングを実行します。この実証例では、サンプリング・ポイントを ARES に設定し、サンプリング・パターンのマッチング条件をマッチング条件 2 (現在のサンプルがその前にサンプリングした値と異なる) に設定しています。

パイプの中の水流がプロペラを回転させ、それによって磁石が回転します。磁石の N-S 極がリード・スイッチのどちらかと平行になると、そのスイッチが閉じます。これによりリード・スイッチの出力にロジック 1 が生成されます。このロジックは対応する入力サンプリング GPIO から使用可能です。

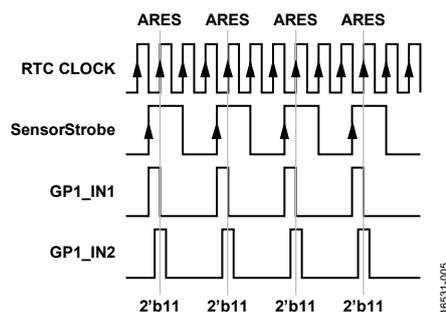


図 5. 2つの GPIO 入力 (リード・スイッチ 1 とリード・スイッチ 2) による入力サンプリング

最初に水流を計算する場合、サンプリング・パターンのマッチング条件はマッチング条件 3 (現在のサンプルが予想サンプル値と一致する) に設定され、予想サンプル値は 2'b11 にセットされています。現在のサンプル値と予想サンプル値が一致した場合、例えば  $SMPx_t = EXP\_SMPx = 2'b11$  の場合に、MCU は、ウェークアップして 2'b11 のカウント数をインクリメントしてから、休止モードに入ります。このプロセスは 1 秒間継続し、MCU は GP1\_IN1 および GP1\_IN2 (表 2 参照) の 2'b11 のパルス数をカウントします。この 2 本のピンでの 2'b11 のパルス数を使用して、リード・スイッチの入力パルスの周波数を計算します。この周波数は、間接的に磁石とプロペラの回転数を表します (式 1 参照)。SensorStrobe 周波数は、計算された磁石回転数の最新値の整数倍に変更されます。

$$f = 1 \div 2'b11 \text{ のパルス数} \quad (1)$$

最初の磁石回転数を計算すると、サンプリング・パターンのマッチング条件は、マッチング条件 2 (現在のサンプルがその前にサンプリングした値と異なる) に変更されます。SensorStrobe の周波数が、例えばサンプリング周波数に変更されると、入力サンプリングによって SensorStrobe パルスの ARES で収集されるデータは、その前のサンプリング・データと同じになります。サンプリング・パターンのマッチング条件により、MCU は現在のサンプル値がその前のサンプル値と異なる場合にのみウェークアップします。マッチング条件を満たした場合、MCU はウェークアップして GP1\_IN1 および GP1\_IN2 ピン (表 2 参照) から磁石の新しい回転数を計算します。次いで、SensorStrobe の周波数を新しい計算値に設定した後、休止モードに入ります。また、これは管内の水の流速が変化することを意味するので、システムは、課金処理のために必要な水の消費量を計測するために水の新しい流速を計算する必要があります。MCU は、水の流速が変化したときにウェークアップするか介入するだけなので、MCU の介入時間と消費電力を削減できます。

入力サンプリングのソフトウェア・フロー

図 6 に入力サンプリングのシステム例におけるソフトウェア・フローを示します。

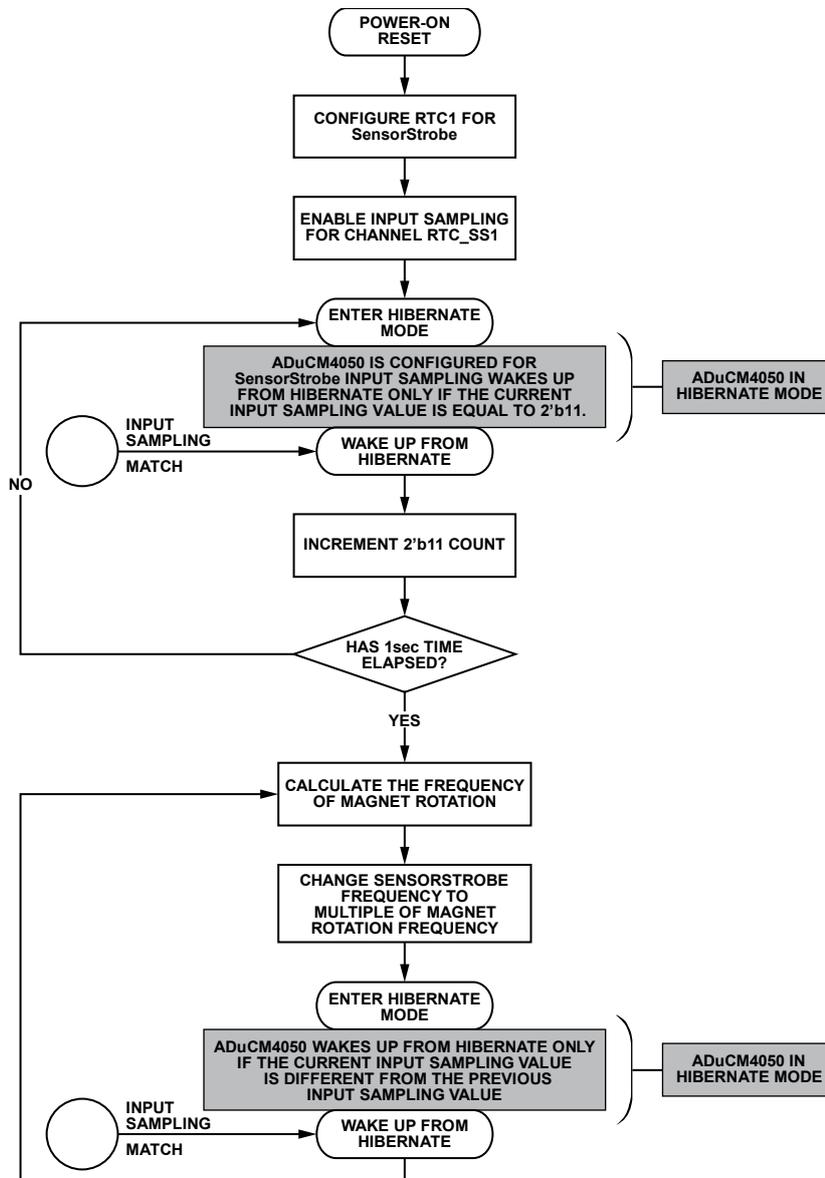


図 6. 低消費電力の水量計システム例の入力サンプリングのフロー・チャート

16531-006

## ソース・コードの構成

図7と図8に、SensorStrobe用および入力サンプリング用に RTC1 を設定する場合のソース・コードを示します。

```

////////////////////////////////////
//////////////////////////////////// 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)

```

図7. RTC1 を SensorStrobe 用に設定するソース・コード

```

////////////////////////////////////
//////////////////////////////////// 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)

```

図8. RTC1 を入力サンプリング用に設定するソース・コード

## 電力測定

ここでは、システムの電流消費量をモニタする方法を説明します。図 1 に示す、電流測定用デジタル・マルチメータ (DMM) のシステム接続を参照してください。

### SensorStrobe のモニタリング

1. EV-COG-AD4050LZ 開発ボードの SensorStrobe アプリケーションをロードします。
2. TH2 のシャント・ジャンパを外します。
3. DMM の正端子を TH2 の左側の端子に接続し、DMM の負端子を TH2 の右側の端子に接続します。
4. 開発ボードの RESET ボタンを押します。
5. DMM の消費電流をモニタします。

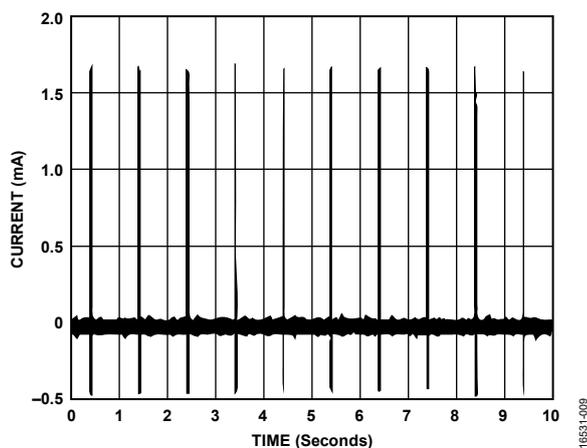


図 9. ADXL362 を接続して動作する SensorStrobe の電流プロファイル

### 入力サンプリングのモニタリング

1. EV-COG-AD4050LZ の低消費電力水量計アプリケーションをロードします。
2. TH2 のシャント・ジャンパを外します。
3. DMM の正端子を TH2 の左側の端子に接続し、DMM の負端子を TH2 の右側の端子に接続します。
4. 開発ボードの RESET ボタンを押します。
5. DMM の消費電流をモニタします。

低消費電力水量計アプリケーションで電流を測定する間、パイプの水流が 5 秒ごとに変化しているため、そのたびに ADuCM4050 はウェークアップして流量測定を実行します (図 10 参照)。

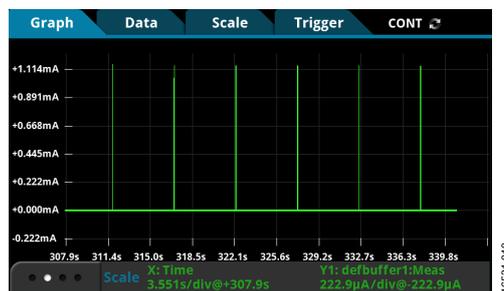


図 10. 低消費電力水量計アプリケーションによって動作する入力サンプリングの電流プロファイル