参考資料

TI Designs: TIDM-1002

MSP430™を使用する超音波ガス・メータのフロントエンドのリファレンス・デザイン

概要

TIDM-1002は、設計者が独自の専用アルゴリズムと、アナログ/デジタル・コンバータ(ADC)ベースのサンプリング技法を使用して、超音波ガス計量サブシステムを開発するために役立ちます。TIDM-1002は256KB FRAMマイクロコントローラ(MCU)のMSP430FR5994をベースとし、LEA (Low Energy Accelerator)とディスクリート・アナログ・フロントエンド(AFE)チップセットを内蔵して、低消費電力で優れた計測能力を実現します。

リソース

TIDM-1002	デザイン・フォルダ
MSP430FR5994	プロダクト・フォルダ
OPA836	プロダクト・フォルダ
OPA835	プロダクト・フォルダ
TS5A9411	プロダクト・フォルダ

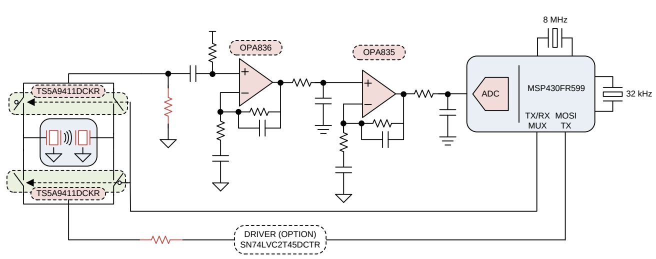
TI E2E™ Community　E2Eエキスパートに質問

特長

- クラス最高の計測能力: ゼロ・フロー・ドリフト1ns未満、シングル・ショット標準偏差2ns未満
- 低消費電力: 400kHzトランスデューサおよび毎秒1サンプルで50µA未満
- 各種のメータとトランスデューサをテストできる柔軟性
- PCのグラフィカル・ユーザー・インターフェイス(GUI)を使用して簡単にテストおよびカスタマイズ可能
- LaunchPad™開発キットのエコシステムと互換

アプリケーション

- ガス計量



1 System Description

The TIDM-1002 is built using TI's MSP430FR5994 MCU, TI's operational amplifiers OPA835 and OPA836, TI's analog switch TS5A9411, and other discrete analog components.

The implementation is based on the calculation of differential time of flight (TOF) involving two transducers for upstream and downstream paths. Signal captures are implemented using the MSP430FR5994's internal, 12-bit ADC and a proprietary technique to increase the sample rate. The signal is then passed through a series of algorithms using MSP430™ MCU's LEA to calculate the necessary output data in a quick and power-effective manner.

The TIDM-1002 is compatible with TI's Launchpad ecosystem by using the MSP-EXP430FR5994 LaunchPad Development Kit and an ultrasonic AFE BoosterPack™ Plug-in Module. The ultrasonic AFE BoosterPack contains all the necessary circuitry to interface with different types of ultrasonic transducers, which include voltage-feedback operational amplifiers, analog switches, and optional voltage converters and level shifters. All hardware files required to recreate this design are provided.

The software is written in a modular and portable manner by using TI's MSPWare and MSP430's Ultrasonic Sensing Gas Metering Library.

The reference design also includes a PC GUI, which enables designers to modify and optimize different configuration parameters. The PC GUI allows for an easier implementation and customization of different transceivers without software modifications.

Source code for an application example and corresponding CCS and IAR projects are provided.

1.1 Key System Specifications

表 1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Single-shot standard deviation (STD)	<2 ns	See 3.2.2.1
Zero-flow drift (ZFD)	<1 ns	See 3.2.2.2
Minimum detectable flow (MDF)	7.95 lph	See 3.2.2.3
Average current consumption at one measurement per second (P_{AVG})	<50 µA	See 3.2.2.4

2 System Overview

2.1 Block Diagram

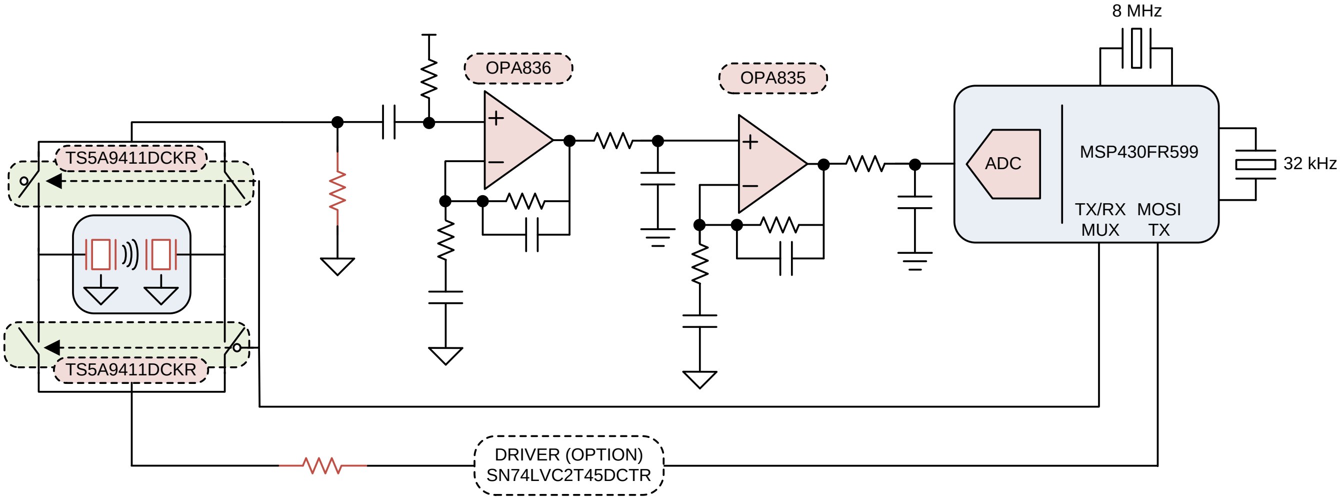


図 1. Ultrasonic Gas Meter Front End Block Diagram

2.2 Design Considerations

2.2.1 Flow Measurement

The ultrasonic gas flow meter front-end design is based on the principle of TOF measurement. This measurement refers to the time taken for a signal to travel from a transmitting transducer to a receiver transducer.

Consider 図 2, which represents a gas pipe.

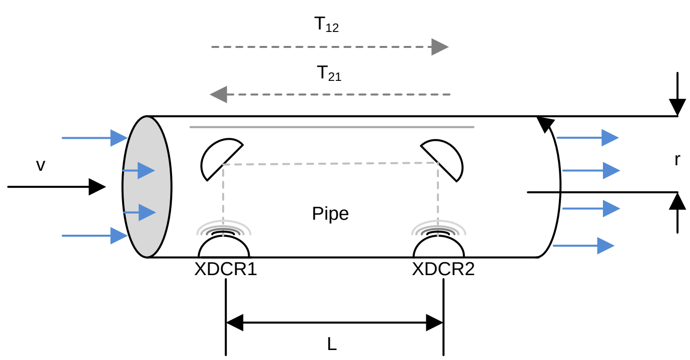


図 2. TOF in Flow Meter

The propagation time of a signal traveling from the first transducer, XDR_1, to a second transducer, XDR_2, is given by T_{12}. In the same way, T_{21} represents the propagation time in the opposite direction. These timings can be calculated according to the following equations:

$$T_{12} = \frac{L}{c + v} \quad (1)$$

$$T_{21} = \frac{L}{c - v} \quad (2)$$

$$\Delta t = T_{21} - T_{12} \quad (3)$$

where:

- c is the velocity of ultrasound in the medium
- v is the velocity of gas flow
- L is the propagation length of the pipe

Using 式 1 through 式 3, the velocity of gas flow (v) can be derived with or without knowing the velocity of ultrasound in the medium (c).

This reference design calculates gas flow assuming that the velocity of ultrasound is unknown. In this case 式 4 can be derived from 式 1 and 式 2.

$$v = \frac{L}{2} \times \left(\frac{1}{T_{12}} - \frac{1}{T_{21}} \right) = \frac{L}{2} \times \left(\frac{T_{21} - T_{12}}{T_{12}T_{21}} \right) = \frac{L}{2} \times \left(\frac{\Delta t}{T_{12}T_{21}} \right) \quad (4)$$

Based on 式 4, the propagation times, T_{12} and T_{21}, must be calculated.

2.2.2 TOF Measurement

In the implementation discussed in this reference design, the MSP430FR5994 performs the complete acquisition process using only external discrete components for signal conditioning. 図 3 shows this process.

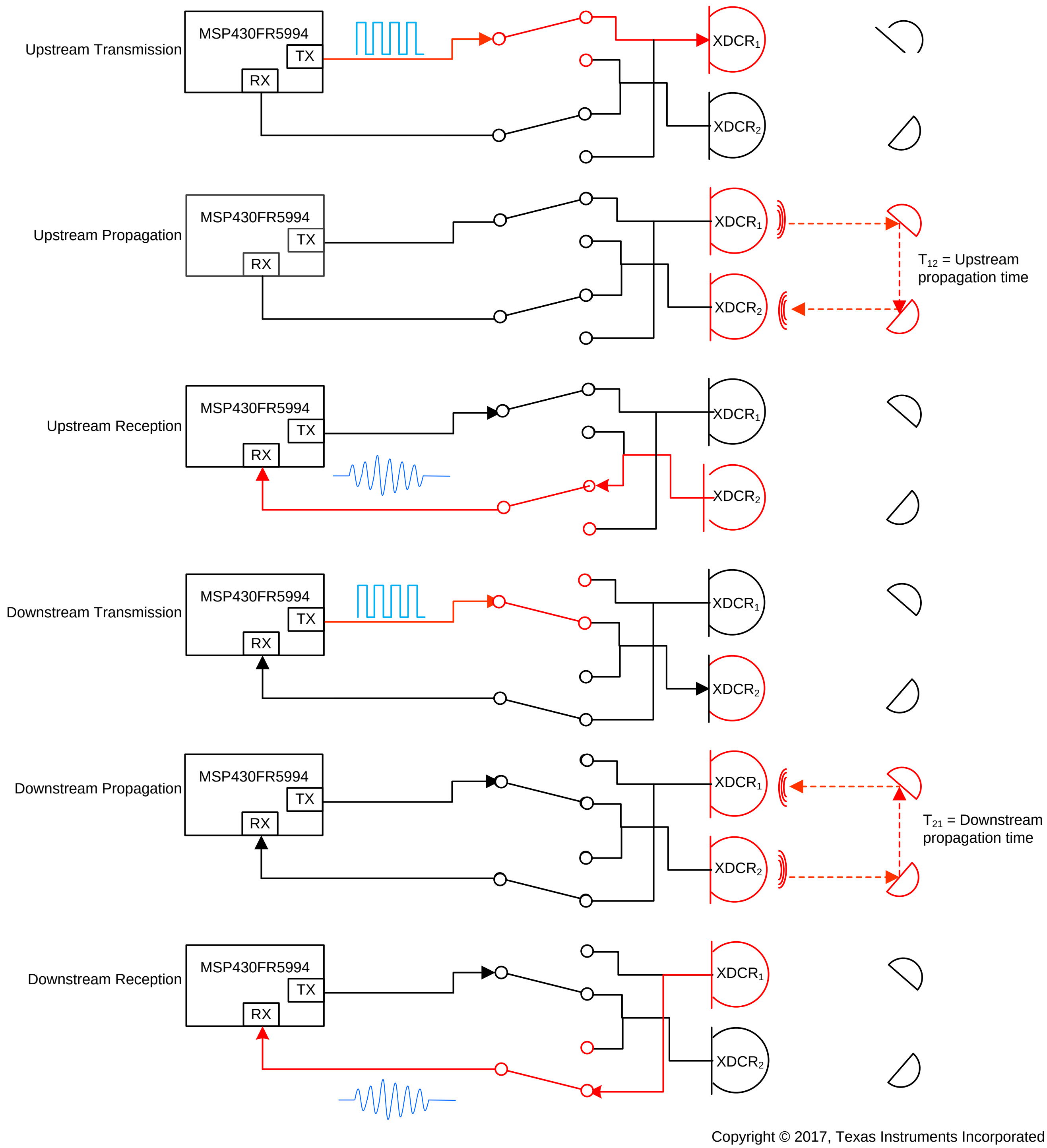


図 3. Ultrasonic TOF Measurement

At the beginning of the sequence, the MSP430FR5994 sends a train of pulses to the first transducer, $XDCR_1$. The signal is then received by the second transceiver, $XDCR_2$, after a propagation time, T_{12} . The difference in time between the transmission and reception determines the upstream TOF, TOF_{UPS}.

The same process is repeated in the opposite direction during the downstream stage resulting in the propagation time, T_{21}, which represents the downstream TOF, TOF_{DNS} .

The differential TOF, Δt, can then be calculated as the difference between T_{12} and T_{21} as described in 式 3.

The differential TOF is typically measured using two techniques:

- Zero-crossing using time-to-digital converter (TDC)
- Correlation using ADC

The TIDM-1002 uses the ADC-based technique because of the following advantages over the TDC technique:

1. Improved performance. The correlation acts as a digital filter to suppress noise, which results in a benefit of approximately three to four times lower noise standard deviation. Similarly, the correction filter also suppresses other interference like line noise.
2. Improved robustness to signal amplitude variations because the algorithm is insensitive to the received signal amplitude, transducer-to-transducer variation, and temperature variation.
3. The envelope of the signal is naturally obtained enabling tuning to the transducer frequencies and detection of envelope variation across time, which can be used for detection of aging of transducers or meters.

2.2.2.1 ADC-Based Acquisition Process

The ADC-based acquisition process implemented in this reference design makes heavy use of the MSP430FR5994's peripherals and hardware capabilities to completely automate the sampling process. This not only provides a tighter control of the sampling process without dependencies on CPU latencies and compilers, but it also reduces the power consumption since the CPU is sleeping during the measurement.

図 4 shows a timing diagram of the signal acquisition process.

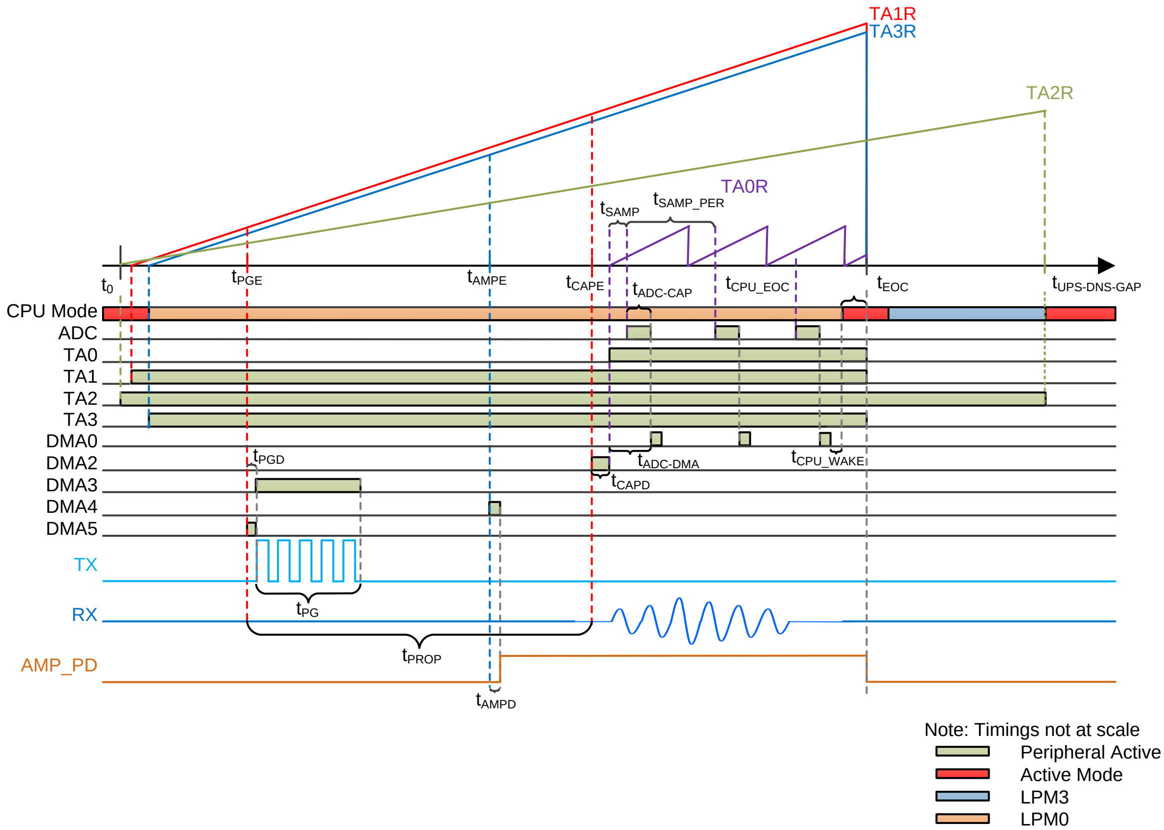


図 4. Signal Acquisition Process for MSP430FR5994

The signal acquisition steps are:

1. At the start of the process (t_0) the device initializes three timers: TA2 running from ACLK, which is expected to trigger the end of the sequence; TA1 running from SMCLK, which is used to trigger the start of pulse generation and ADC sampling; and TA3 also running from SMCLK, which is used to enable the amplifiers.
2. The device goes to LPM0 at this moment waiting for the sequence to finish.
3. The pulse generation event (t_{PGE}) generated by TA1.0 triggers DMA5.
4. DMA5 performs a single transfer to enable DMA3 to generate pulses using SPI after a DMA transfer delay (t_{PGD}).
5. DMA3 generates pulses autonomously for the duration of the pulse generation (t_{PG}), which depends on the specified transducer frequency and number of samples.
6. The amplifier event (t_{AMPE}) generated by TA3.0 triggers DMA4.
7. After a DMA transfer delay (t_{AMPD}), DMA4 enables the amplifiers by writing the GPIO AMP_PD.
8. The capture event (t_{CAPE}) generated by TA1.2 triggers DMA2. Note that the delay between the pulse generation event (t_{PGE}) and the capture event (t_{CAPE}) is expected to be the propagation delay (t_{PROP}).
9. After a DMA transfer delay (t_{CAPD}), DMA2 performs a single transfer enabling TA0 to generate a sampling trigger for the ADC.
10. TA0 settings configure the sample and hold timing of the ADC (t_{SAMP}) as well as the sampling period (t_{SAMP_PER}).
11. On every trigger from TA0, the ADC starts a conversion taking a predefined number of cycles ($t_{ADC\text{-}CAP}$) depending on bit resolution and ADC clock source. After every conversion the ADC triggers DMA0 to automatically store the results in memory.
12. The process is repeated for the specified number of samples. 図 4 only shows three samples for simplicity purposes.

13. After the last sample, DMA0 will automatically wake up the CPU after a wake-up delay (t_{CPU_WAKE}).
14. The CPU wakes up and configures all peripherals to their idle state at the end of conversion after some cycles (t_{CPU_EOC}).
15. All peripherals are in an idle state and the CPU prepares to go to LPM3 at the end of conversion time (t_{EOC}).
16. After a specified ups-dns gap ($t_{UPS-DNS-GAP}$) where the lines are expected to settle, TA2.0 triggers the CPU to wake-up and either restarts the process for the other channel or processes the data.

Due to the critical timing of each one of these events, it is advised to avoid using interrupts or other DMA channels during the sampling process. Synchronous events can result in unexpected latencies while asynchronous events can also result in unwanted discrepancies from sample to sample.

2.2.2.2 Signal Processing and Ultrasonic Sensing Gas Metering Library

This reference design is intended to explain some of the key concepts for implementing an ultrasonic, gas metering front end, but it is important to note that the proposed solution makes use of the Ultrasonic Sensing Gas Metering Library, which implements several of TI's proprietary algorithms to capture and process the signal as well as calculates the TOF and air flow rate.

The Ultrasonic Sensing Gas Metering Library and its documentation are available at GasLibrary.

2.2.3 Low-Power Design

The front end presented in this document was designed to meet the low-power requirements from gas meter applications. While the current consumption of the system depends on the meter, the transducers, and the configuration of the application, the solution was designed to consume less than 50 µA using 400-kHz transducers taking one measure per second. This includes one upstream and one downstream measurement per second and the associated signal processing.

In some configurations, the system can consume less than 20 µA. More details on power consumption measurements can be found in 3.2.2.4.

The low-power features implemented by this application include:

- Energy-efficient software
- Optimized hardware design
- Efficient use of FRAM
- The LEA advantage

2.2.3.1 Energy-Efficient Software

The application software and Ultrasonic Sensing Gas Metering Library used by the TIDM-1002 maximize the use of low-power modes. The MCU and external circuitry will go to the lowest-power mode available when possible.

2.2.3.2 Optimized Hardware Design

The TIDM-1002 not only uses the MSP430 platform, which combines high performance with industry leading ultra-low-power (ULP) consumption, but all external components were also selected for their energy efficiency.

2.2.3.3 Efficient Use of FRAM

The MSP430FR5994 uses FRAM technology, which combines the best of Flash and RAM. FRAM offers the non-volatility of Flash together with fast and low-power writes, write endurance of 10^{15} cycles, resistance to radiation and electromagnetic field, and unmatched flexibility.

This use of technology results in an application that writes and logs data more efficiently than its Flash-based counterparts.

The application provided in this software package makes use of FRAM to store several non-volatile variables, such as the configuration received from the GUI. This storage allows users to reconfigure the device on-the-fly and keeps the configuration after subsequent power cycles.

Developers of metering applications can make use of FRAM to implement many additional features, including:

- Log historical statistical data, such as differential TOF, absolute TOF, or volume
- Log errors and faults in non-volatile-memory
- Save and restore the state of the device before a power failure (see TIDM-FRAM-CTPL)

2.2.3.4 The LEA Advantage

The TIDM-1002 also makes efficient use of the LEA available in the MSP430FR5994.

LEA is a 32-bit hardware engine designed to perform signal processing, matrix multiplications, and other operations that involve vector-based signal processing, such as FIR, IIR, and FFT without CPU intervention. Efficient use of this module can result in improvements of up to 36.4 times for typical math intensive operations.

LEA is heavily used by the Ultrasonic Sensing Gas Metering Library to accelerate all vector operations in its proprietary algorithms. LEA not only reduces the processing time of many operations, but LEA also allows the CPU to go to a low-power state. During the implementation of this reference design, LEA demonstrated the improvement outlined in 表 2 in performance and power consumption.

表 2. Processing and Power Consumption With and Without LEA

PARAMETER	USING LEA	NOT USING LEA
GASMETERING_runAlgorithm()execution time	12.8 ms	93.7 ms
Current consumption attributed to algorithms over one sample per second (MCLK = 8 MHz)	19.78 µA	144 µA

For more information about LEA, see *Low-Energy Accelerator (LEA) Frequently Asked Questions (FAQ)* [1] and *Benchmarking the Signal Processing Capabilities of the Low-Energy Accelerator on MSP430 MCUs* [2].

2.3 Highlighted Products

2.3.1 MSP430FR5994

The MSP430FR599x MCUs take low power and performance to the next level with the unique LEA for digital signal processing (DSP). This accelerator delivers 40 times the performance of ARM® Cortex®-M0+ MCUs to help developers efficiently process data using complex functions, such as FFT, FIR, and matrix multiplication. Implementation requires no DSP expertise with a free optimized DSP library available. Additionally, with up to 256KB of unified memory with FRAM, these devices offer more space for advanced applications and flexibility for effortless implementation of over-the-air firmware updates.

The MSP ULP FRAM MCU platform combines uniquely embedded FRAM and a holistic ULP system architecture, which allows system designers to increase performance while lowering energy consumption. FRAM technology combines the low-energy fast writes, flexibility, and endurance of RAM with the non-volatile behavior of Flash.

The MSP430FR5994 includes a powerful and flexible set of peripherals: four eUSCI_A to implement UART, IrDA or SPI; four eUSCI_B to implement I²C or SPI; a 12-bit ADC with up to 20 external channels; an analog comparator with up to 16 channels; six 16-bit timers; an AES accelerator; and up to 68 IOs.

The ADC-based acquisition process implemented in this reference design makes heavy use of the MSP430FR5994’s peripherals and hardware capabilities to completely automate the sampling process. These capabilities not only provides a tighter control of the sampling process without dependencies on CPU latencies and compilers but also reduces the power consumption because the CPU is sleeping during the measurement.

The TIDM-1002 uses the MSP430FR5994 not only to act as a host processor communicating with a PC GUI but also to perform measurements in an automated process. The powerful peripherals of the MSP430FR5994, together with FRAM technology and the LEA, allow for an accurate and efficient implementation of an ultrasonic gas meter front end.

2.3.2 OPA835 and OPA836

The OPA835 and OPA836 are single, ultra-low power, rail-to-rail output, negative-rail input, voltage-feedback (VFB) operational amplifiers designed to operate over a power-supply range of 2.5 V to 5.5 V with a single supply or ±1.25 V to ±2.75 V with a dual supply. Consuming only 250 µA (OPA835) or 1 mA (OPA836) per channel and a unity-gain bandwidth of 56 MHz (OPA835) or 205 MHz (OPA836), these amplifiers set an industry-leading, power-to-performance ratio for rail to-rail amplifiers.

Coupled with a power-savings mode to reduce current to <1.5 µA, these devices offer an attractive solution for high-frequency amplifiers in power-sensitive applications.

The TIDM-1002 uses an OPA835 and an OPA836 to implement an efficient two-stage amplifier providing the desired gain and bandwidth.

A high gain first stage amplifier is implemented using an OPA836 due to the following reasons:

- Low input noise (4.6 nV/√Hz at 100 KHz)
- High bandwidth (205 MHz),
- Low-power consumption: 0.5 µA in power-down mode and 1-mA quiescent current.

Because the gain of the second stage amplifier is lower than the first one, the low input noise and bandwidth requirements are not as critical. The OPA835 was selected due its lower quiescent current of 250 µA.

2.3.3 TS5A9411

The TS5A9411 device is a bidirectional, single-pole double-throw (SPDT) analog switch that is designed to operate from 2.25 V to 5.5 V. The device offers low ON-state resistance, low leakage, and low power with a break-before-make feature. These features make this device suitable for portable and power-sensitive applications.

The TIDM-1002 uses the TS5A9411 to switch the transmission and reception signals from the MCU to the two transceivers in an efficient manner and without distortion.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

3.1.1 Hardware

The hardware used for this reference design consists of the MSP-EXP430FR5994 LaunchPad and an ultrasonic AFE BoosterPack.

3.1.1.1 MSP-EXP430FR5994

The MSP-EXP430FR5994 LaunchPad Development Kit is an easy-to-use evaluation module (EVM) for the MSP430FR5994 MCU. The EVM contains everything needed to start developing on the ULP MSP430FR5x FRAM MCU platform, which includes onboard debug probe for programming, debugging, and energy measurements. The board includes buttons and LEDs for quick integration of a simple user interface as well as two unique features: a microSD card slot that allows the user to interface with SD cards and a super capacitor (super cap) that acts like a rechargeable battery that enables standalone applications without an external power supply.

By being compatible with the LaunchPad Development Kit ecosystem, the MSP-EXP430FR5994 can be connected to an extensive list of BoosterPacks providing functionality, such as Audio, RF, LCD, and so forth.

The TIDM-1002 only uses the MSP-EXP430FR5994 connected to the ultrasonic AFE BoosterPack and it is the developer's responsibility to check compatibility with other BoosterPacks.

図 5 shows the pinout of this LaunchPad.

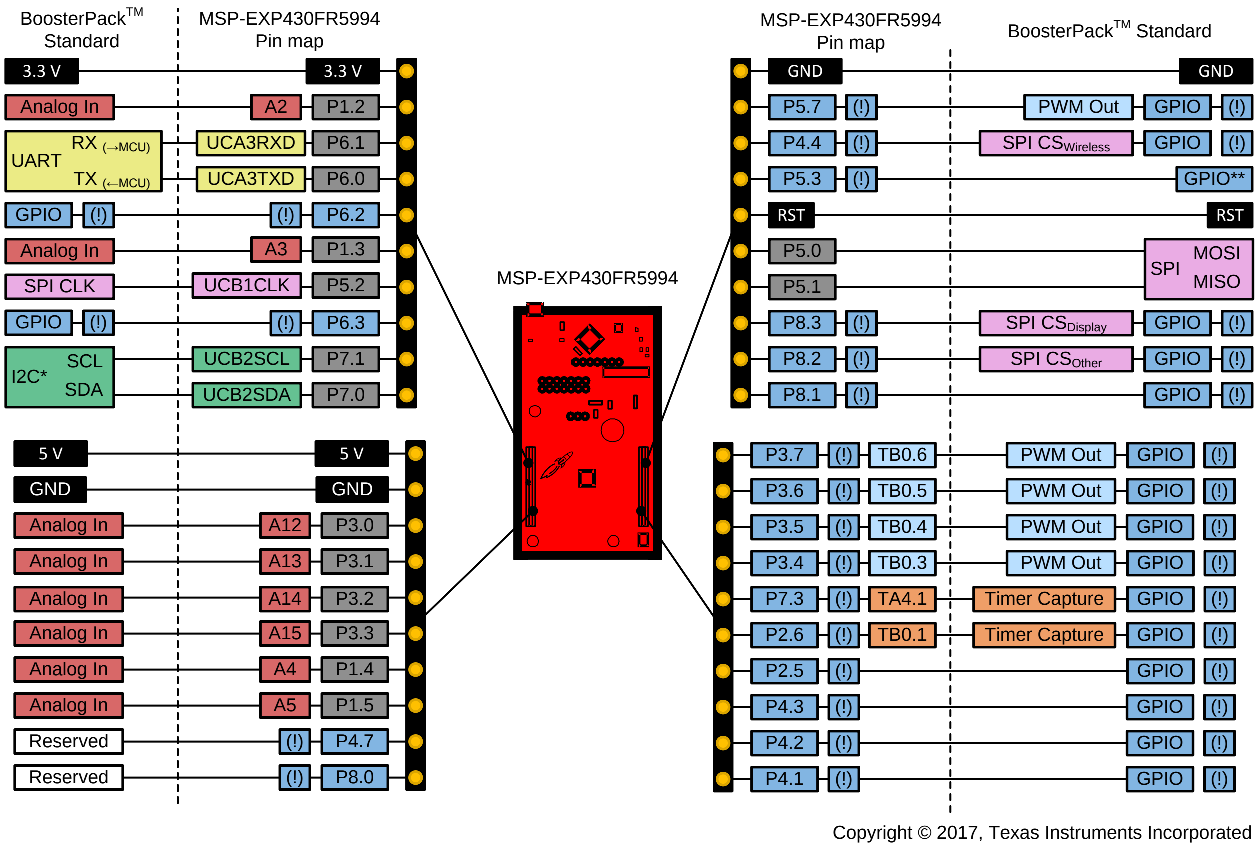


図 5. MSP-EXP430FR5994 Pinout

The MSP-EXP430FR5994 has several jumpers, which are shown in 表 3.

表 3. MSP-EXP430FR5994 Jumper Configuration

JUMPER	DESCRIPTION	PROGRAMMING	EXECUTION (USB POWER)	EXECUTION (EXT POWER)
J101	ON: Connect eZ-FET signals to target area OFF: Disconnect eZ-FET signals to target area	GND: ON 5V: ON 3V3: ON RXD: ON TXD: ON SBWTDIO: ON SBWTCK: ON	GND: ON 5V: ON 3V3: ON RXD: ON TXD: ON SBWTDIO: OFF SBWTCK: OFF	GND: ON 5V: OFF 3V3: OFF RXD: ON TXD: ON SBWTDIO: OFF SBWTCK: OFF
J8	OFF: Don't use super cap to power EVM Use: Use super cap Charge: Charge the super cap	OFF	OFF	OFF
J7	ON: Connect LED1 to P1.0 OFF: Disconnect LED1 from P1.0	ON	ON	ON
J5	Used to provide external power to LaunchPad	Disconnected	Disconnected	Connected to external 3.3 V and GND

3.1.1.2 Ultrasonic AFE Boosterpack™

The ultrasonic AFE BoosterPack was designed to interface the MSP-EXP430FR5994 with common ultrasonic transducers.

図 6 shows the pinout of this BoosterPack.

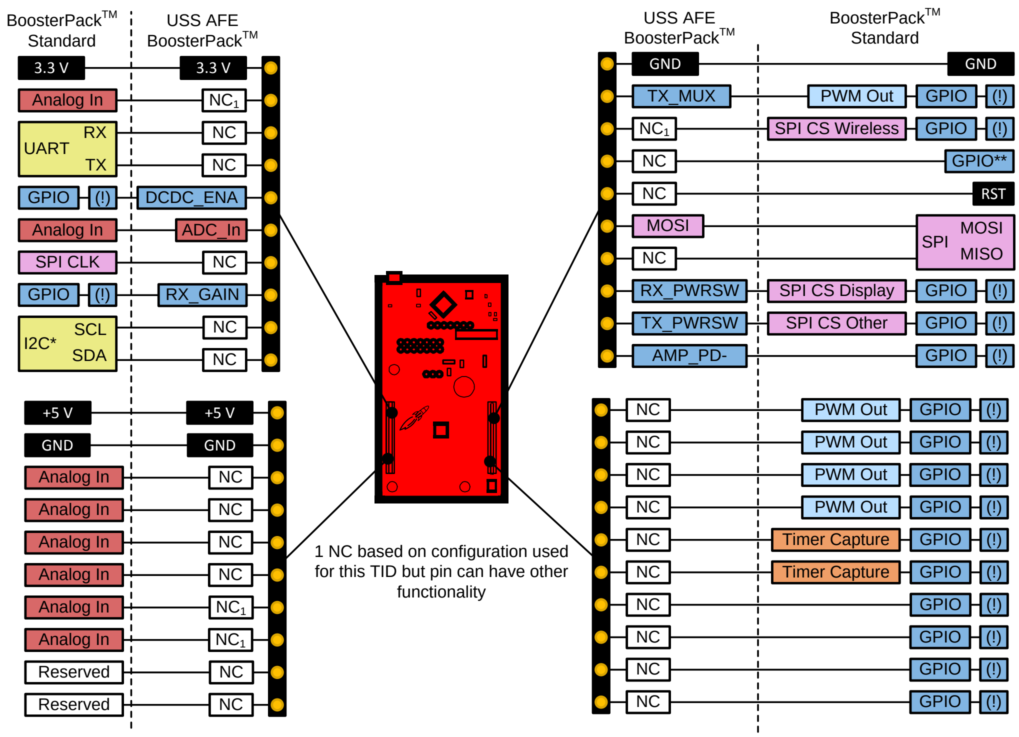


図 6. Ultrasonic AFE BoosterPack™ Pinout

This BoosterPack includes several 0-Ω resistors, jumpers, and connectors that provide the flexibility to use the BoosterPack for different transducers and configurations.

表 4 includes a description of all jumpers and connectors.

表 4. Ultrasonic AFE BoosterPack™ Jumper Configuration

JUMPER	DESCRIPTION	DEFAULT
JP1	ON: Connect 3.3 V to 5-V step-up converter.(1)	OFF
	OFF: Disconnect 3.3 V from 5-V step-up converter.	
	Note: JP4 must be set when JP1 is set.	
JP2	1 to 2: Set TX pulse voltage to 3.3 V.	1-2
	2 to 3: Set TX pulse voltage to 5 V. (1)	
JP3	ON: Use 3.3 V from LaunchPad connector.	ON
	OFF: Disconnect 3.3 V from LaunchPad connector allowing connection of external supply.	
JP4	ON: Enable 5-V step-up converter output.(1)	OFF
	OFF: Disable 5-V step-up converted output.	
	Note: JP1 must be set when JP4 is set.	
J1	Pin1: Transceiver 1 signal	Transceiver 1
	Pin2: Transceiver 1 ground	
J2	Pin1: Transceiver 2 signal	Transceiver 2
	Pin2: Transceiver 2 ground	
J3	Allows for OPA836's current measurement	OFF
J4	Allows for OPA835's current measurement	OFF
J5	GND	GND
J6	GND	GND
J7	Allows for THS4531's current measurement.(1)	OFF

(1) For future use; not supported in this TI Design

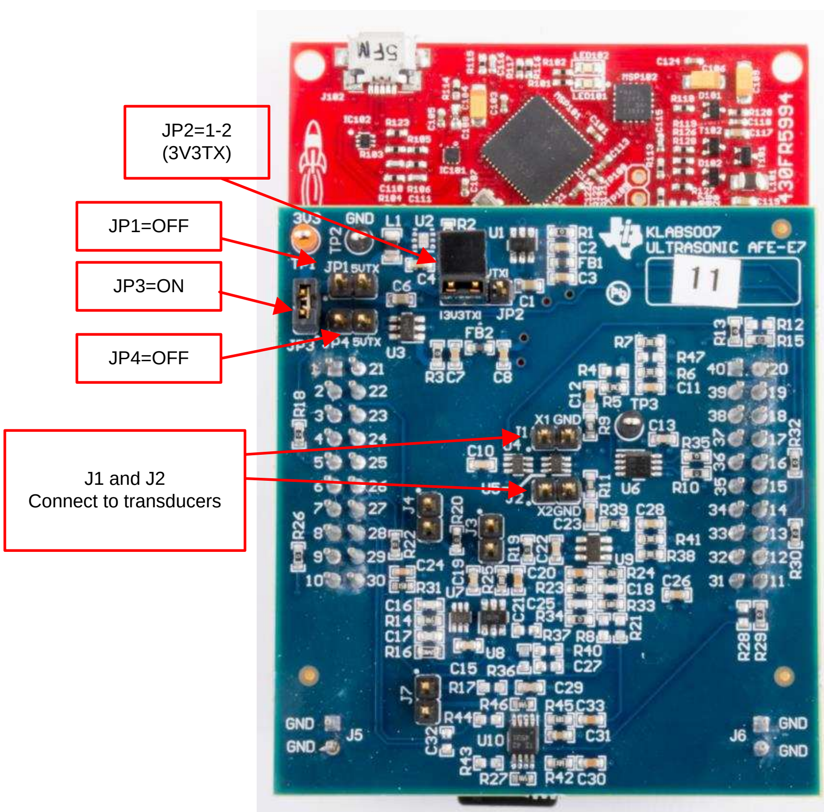


図 7. Default Configuration of Ultrasonic AFE BoosterPack™

The ultrasonic AFE BoosterPack includes features, such as support for 5 V and an extra-differential amplifier, that are only included for future use and are not currently supported.

The description of configurations aside from the default configuration described above is outside the scope of this document, and the testing or implementation of these configurations is not supported by this reference design.

3.1.2 Software

図 8 shows the software architecture implemented in this reference design.

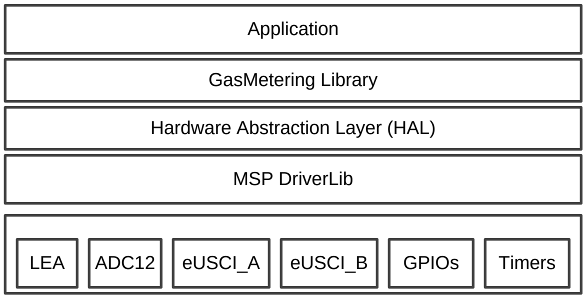


図 8. Software Architecture

The software is implemented in a modular and portable manner; however, this TI Design only includes examples tested on MSP430FR5994 using the MSP-EXP430FR5994 LaunchPad.

The software package includes source code for the application, the Ultrasonic Sensing Gas Metering Library in binary format, and corresponding IAR and Code Composer Studio™ (CCS) projects.

The main software components are discussed in the following sections.

3.1.2.1 MSP Driver Library (MSP DriverLib)

Driver Library (or DriverLib) includes APIs for selected MSP430 device families providing easy-to-use function calls. Each API is thoroughly documented through a user's guide, API Guide, and code examples.

The TIDM-1002 uses the MSP Driver Library to interface with all hardware modules used by the application from the eUSCI_A, which is used for asynchronous communication with the PC, to the ADC12, which is used to sample the incoming signal. This not only allows for an easier migration to other MSP MCUs, but it also makes the code easier to read and understand by using common language APIs.

All DriverLib files used by this application are included in source code in the software package.

MSP DriverLib and documentation are also available at MSPDRIVERLIB.

3.1.2.2 Hardware Abstraction Layer (HAL)

A hardware abstraction layer (HAL) is implemented to encapsulate all hardware interactions allowing for easier customization to different developer hardware requirements and migration between TI platforms.

3.1.2.3 Ultrasonic Sensing Gas Metering Library

The Ultrasonic Sensing Gas Metering Library includes proprietary algorithms to capture and process the signal received from the transceivers as well as to calculate the TOF and air flow.

The library includes an easy-to-implement set of fully-documented APIs that hide the complexity behind ultrasonic measurement calculations and allow for a faster implementation of the application.

A comprehensive list of parameters allows developers to configure the system according to different hardware and transceiver requirements.

The Ultrasonic Sensing Gas Metering Library used by the application is included in CCS and IAR library format.

This library and its documentation are available at *GasLibrary* .

3.1.2.4 *Application*

図 9 shows the flow diagram of the application.

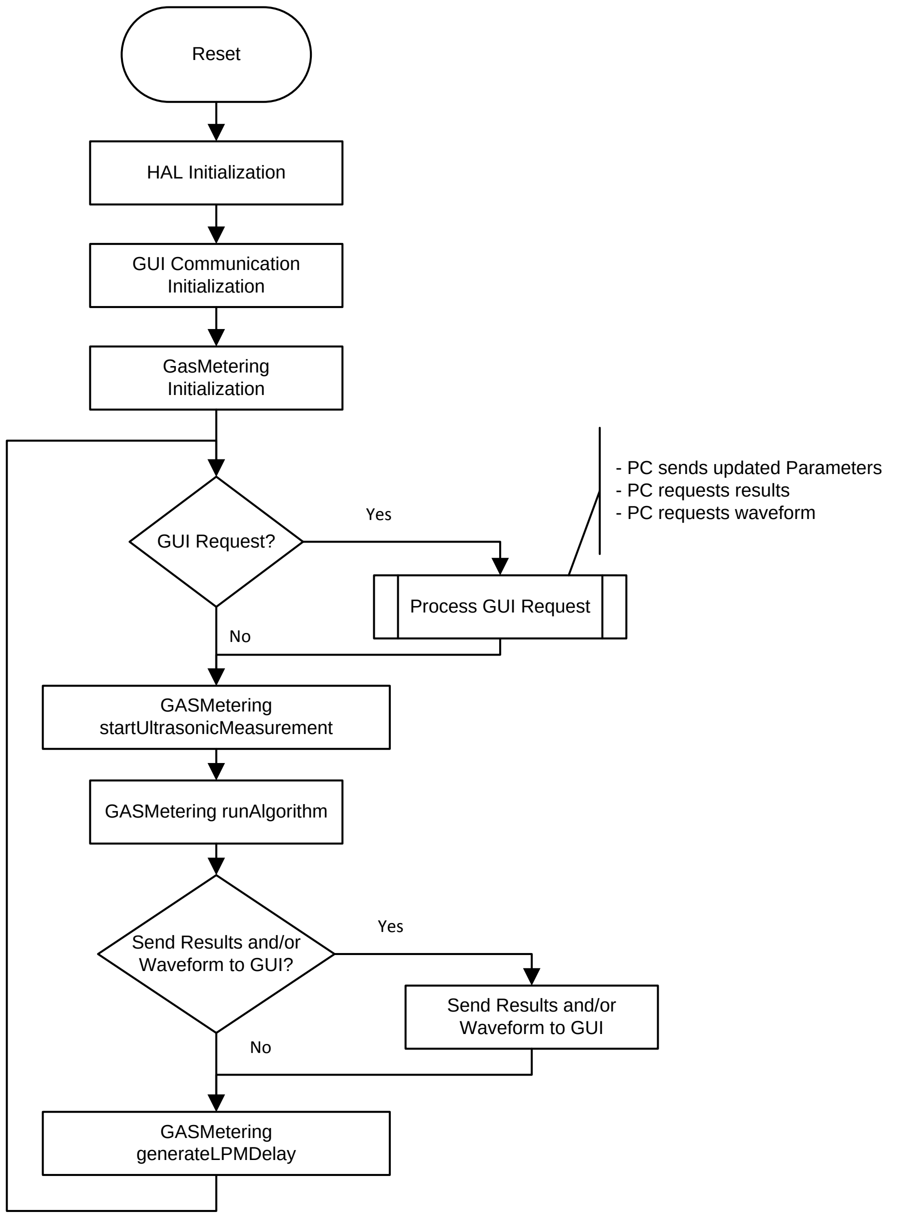


図 9. Ultrasonic Gas Meter Application Flow Diagram

After initialization, the MCU will stay in a continuous loop attending GUI requests, performing measurements, and going into a low power mode when idle.

It is important to note that the device will also go to the applicable low-power mode when possible while doing ultrasonic measurements (for example, while waiting for the propagation time as explained in 2.2.2.1).

3.1.2.5 PC GUI

The MSP430 Ultrasonic Sensor Tool GUI included in this reference design allows developers to modify some configuration parameters required to test different transducers as well as observe the behavior of the system in real time.

The configuration window is shown by default when opening the application. 図 10 shows this window.

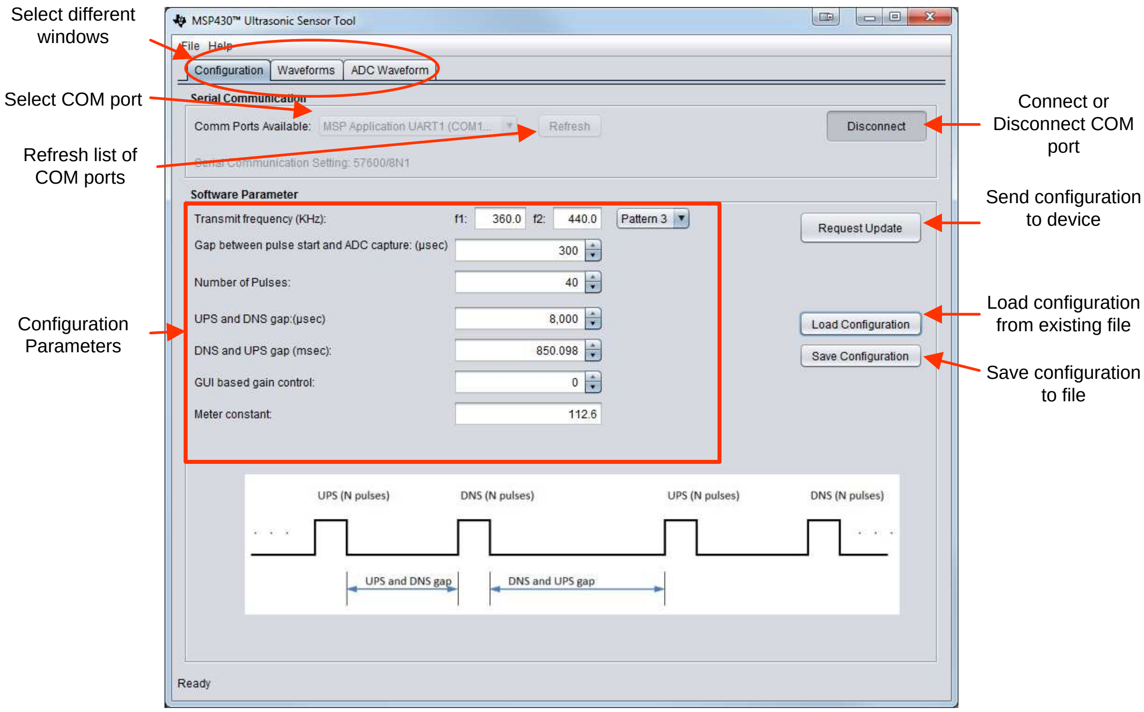


図 10. GUI Configuration Window

The configuration window allows developers to configure the parameters detailed in 表 5.

表 5. Parameters in GUI Configuration Window

PARAMETER	DESCRIPTION
Transmit frequency	When *Pattern 3* is selected, the GUI generates a transmit pattern using frequencies f1 and f2.
Gap between pulse start and ADC capture	Gap in microseconds between the generation of a pulse output and sampling of received signal using ADC
Number of pulses	Number of pulses sent during pulse generation
UPS and DNS gap	Gap in microseconds between upstream and downstream captures
DNS and UPS gap	Gap in milliseconds between captures. This is measured from the end of downstream capture to start of next upstream.
GUI based gain control	Enables control of gain amplifier
Meter constant	Constant used to calculate the calculate volume flow rate. This static constant is related to the cross section area of the meter that vendor should provide as a one time input.

An additional Developer's Configuration window can be opened by pressing F5 or by selecting *Menu → File → Developer's Configuration* (shown in 図 11).

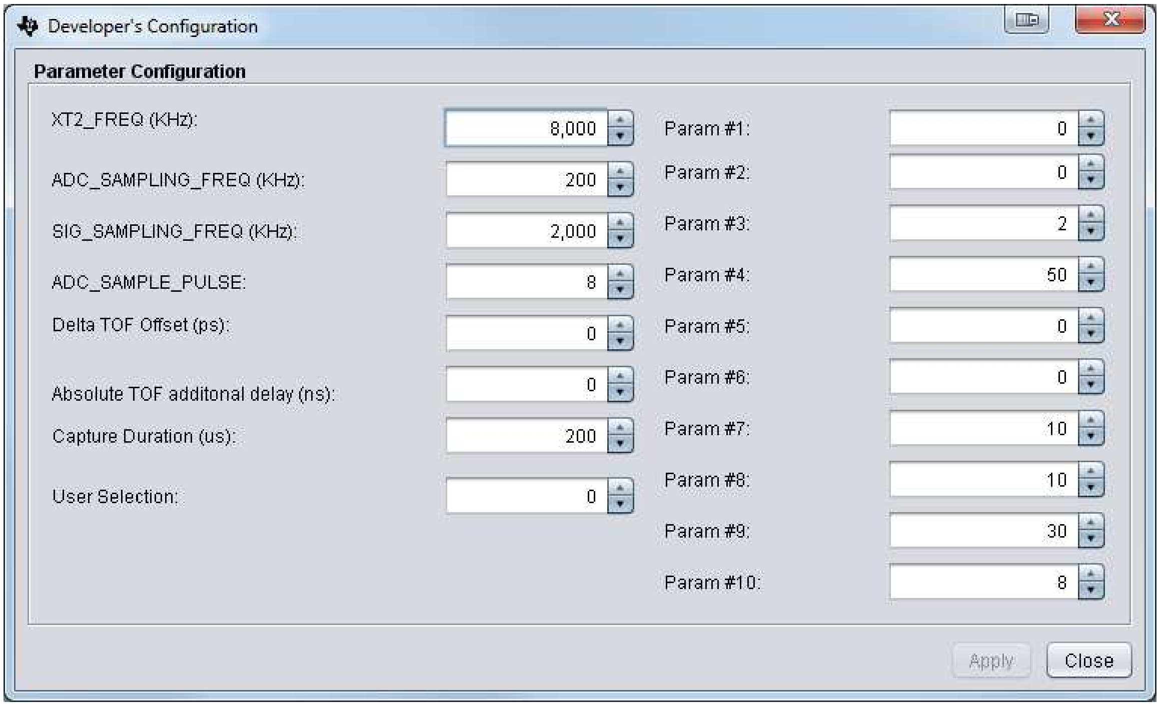


図 11. GUI Developer's Configuration Window

The Developer's Configuration window has the parameters listed in 表 6.

表 6. Parameters in Developer's Configuration Window

PARAMETER	DESCRIPTION
XT2_FREQ	Frequency of XT2 crystal connected to MSP430FR5994 in KHz
ADC_SAMPLING_FREQ	Sampling frequency of the ADC in KHz
SIG_SAMPLING_FREQ	Sampling frequency of the received signal in KHz
ADC_SAMPLE_PULSE	Ignored
Delta TOF offset	Offset to compensate the delta TOF in picoseconds
Absolute TOF additional delay	Time in nanoseconds to compensate for unaccounted additional delays in absolute TOF
Capture duration	Duration of the ADC capture in microseconds
User selection	Ignored
Param #1 (continuous captures)	Used to perform continuous captures of the signal when the value is *100*. Other values are not supported and reserved for future use.
Param #2 (frequency sweep)	Used for transducer/system characterization when the value is *100*. Starting from f1 and ending at f2, a different pattern TX pulse is generated after every UPS and DNS capture in steps of 1KHz. Other values are not supported and reserved for future use.
Param #3 (computation mode)	Computation mode used for algorithms. Refer to the Ultrasonic Sensing Gas Metering Library for more information.
Param #4 (envelope threshold)	Envelope threshold in microseconds used to search for a matching signal
Param #5	Ignored
Param #6 (first amplifier delay)	Adjusts delay of amplifier for first capture
Param #7 (negative range delta TOF)	Maximum negative index to search across correlation value of UPS and DNS for matching signal
Param #8 (positive range delta TOF)	Maximum positive index to search across correlation value of UPS and DNS for matching signal
Param #9 (negative range abs TOF)	Maximum negative index to search across correlation value of UPS and DNS and TX pulse for the matching signal
Param #10 (positive range abs TOF)	Maximum positive index to search across correlation value of UPS and DNS and TX pulse for the matching signal

The MSP430 Ultrasonic Sensor Tool allows developers to observe the behavior of the system in real time by using the *Waveforms* window, shown in 図 12.

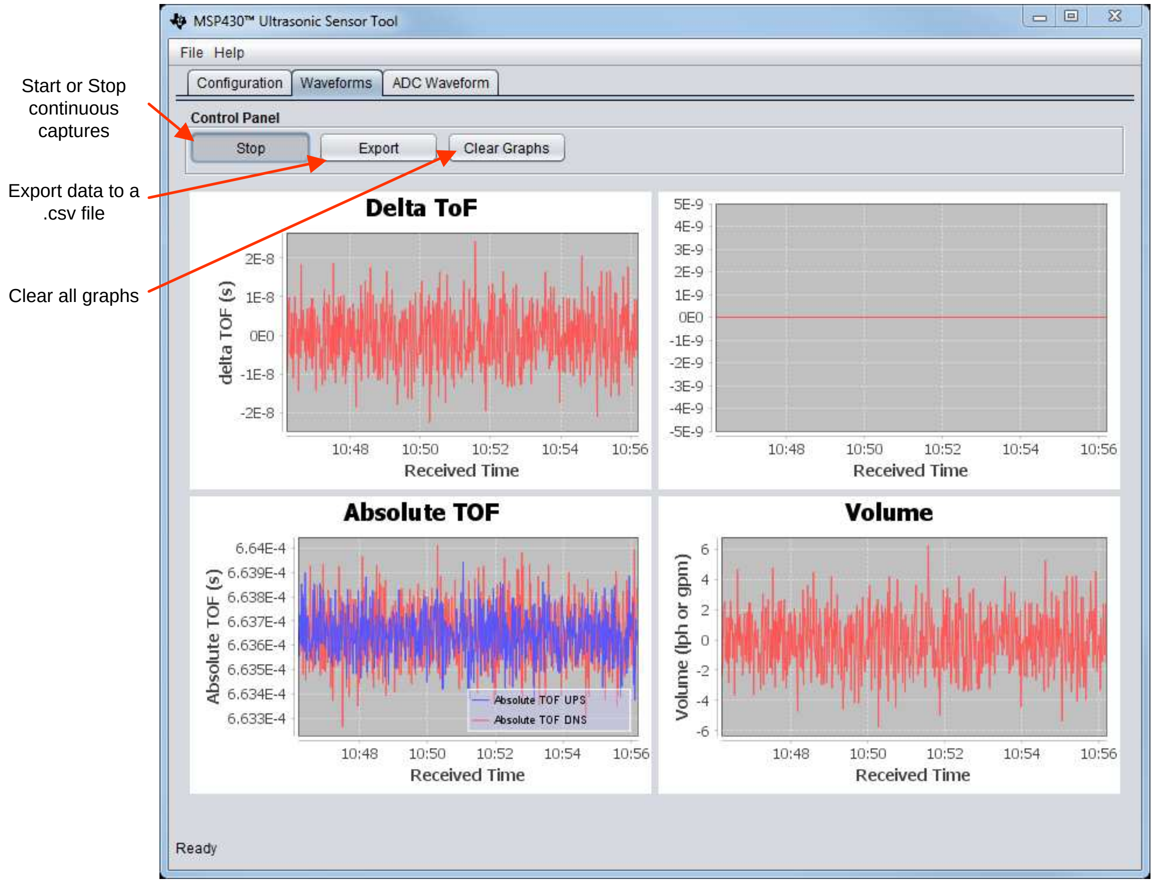


図 12. GUI Waveforms Window

Additionally, it is also possible to obtain and plot a single capture to validate the integrity of the signal by using the ADC Waveform view (図 13).

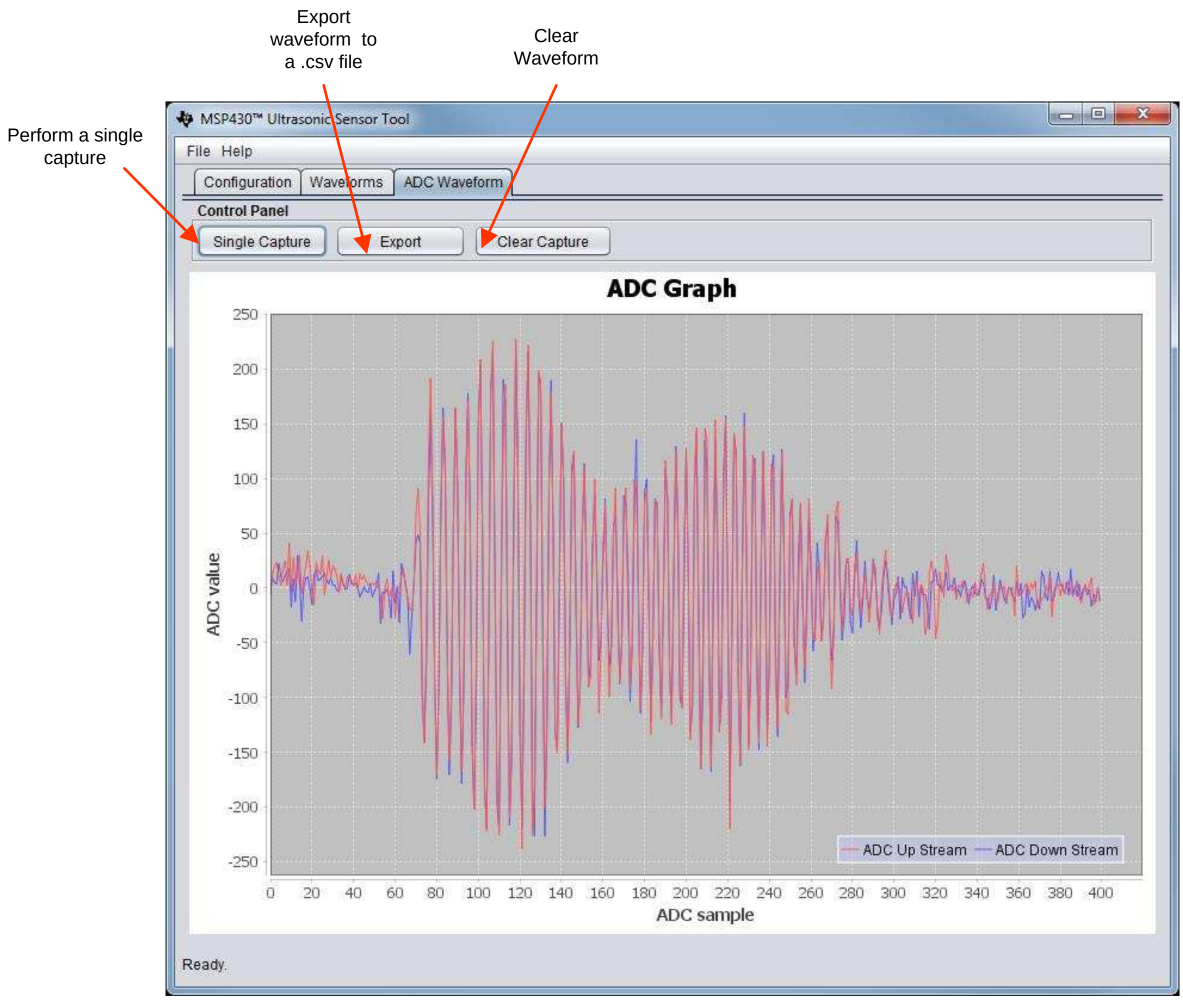


図 13. GUI ADC Waveform Window

3.1.3 Transducer and Meter Implementation

The purpose of this TI Design is to show the implementation of an ultrasonic front end and not the design of a transducer or meter; however, this section shows the implementation of a 3D printed meter, which can be used as a reference to test the system.

The ultrasonic meter that accompanies this TI Design is a 3D-printed tube with CeramTec™ 400-kHz transducers mounted in a *V* configuration, as shown in 図 14. The part number for the 400-kHz transducers is 09300/001. Contact CeramTec directly for more information about their transducers.

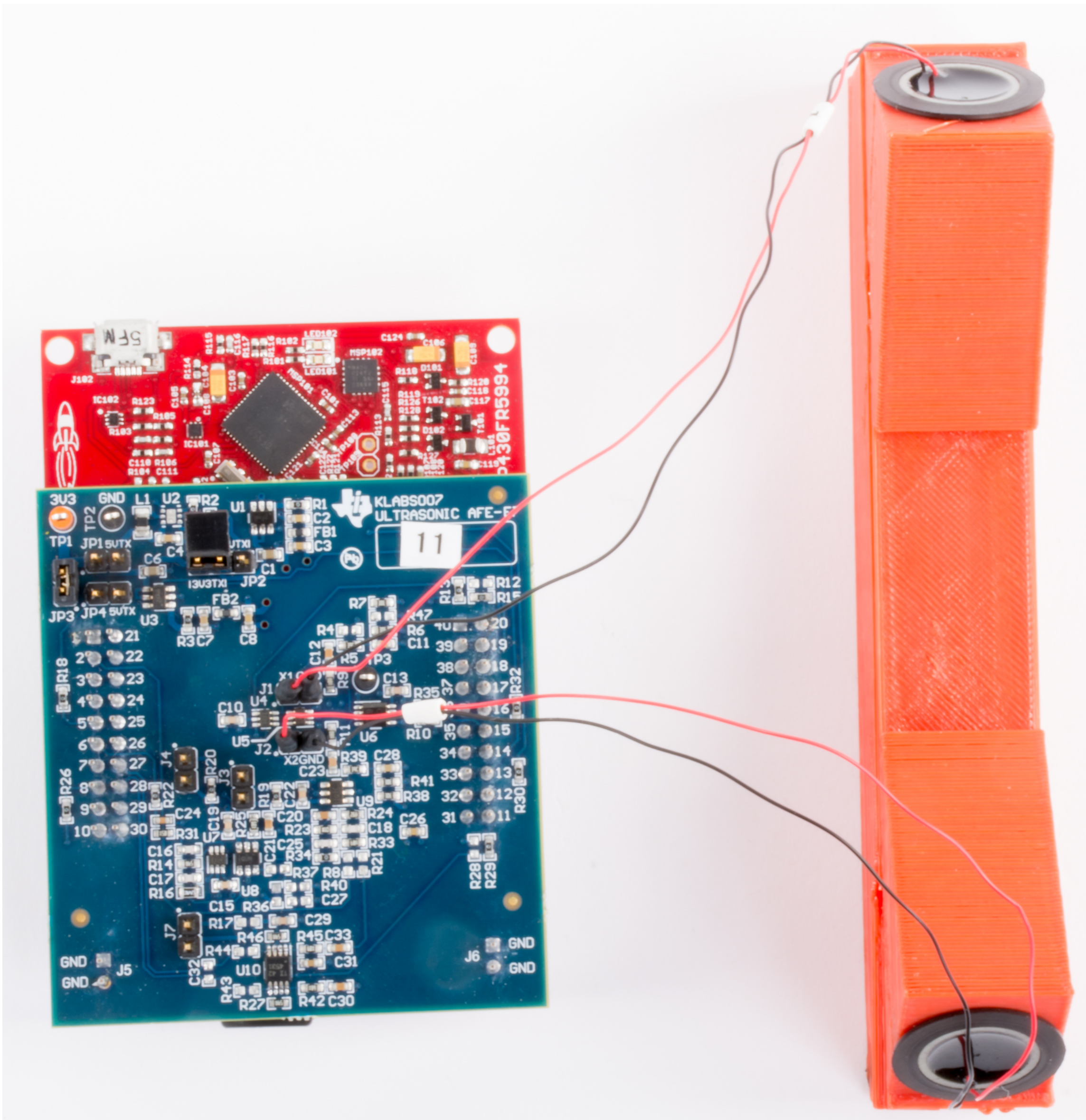


図 14. Ultrasonic AFE BoosterPack™ Connected to V Meter

As explained in 2.2.1, an ultrasonic transmission is first generated by the first transducer and received by the second one. The second transducer then emits an ultrasonic signal that is subsequently received by the first transducer.

A fan can be connected to one of the sides of the meter when doing flow tests. The amount of flow may be varied by changing the voltage to the fan (up to 24V) or by partially covering one end of the flow tube.

The flowtube design is available as an STL file and may be 3D printed with PLA or ABS.

3.1.3.1 Frequency Characterization of Transducer and Meter

Different meters and transducers have different responses; consequently, it is important to characterize the frequency response of the meter to obtain an optimal performance.

The MSP430 Ultrasonic Sensor Tool GUI provided in this TI Design can be used for this purpose.

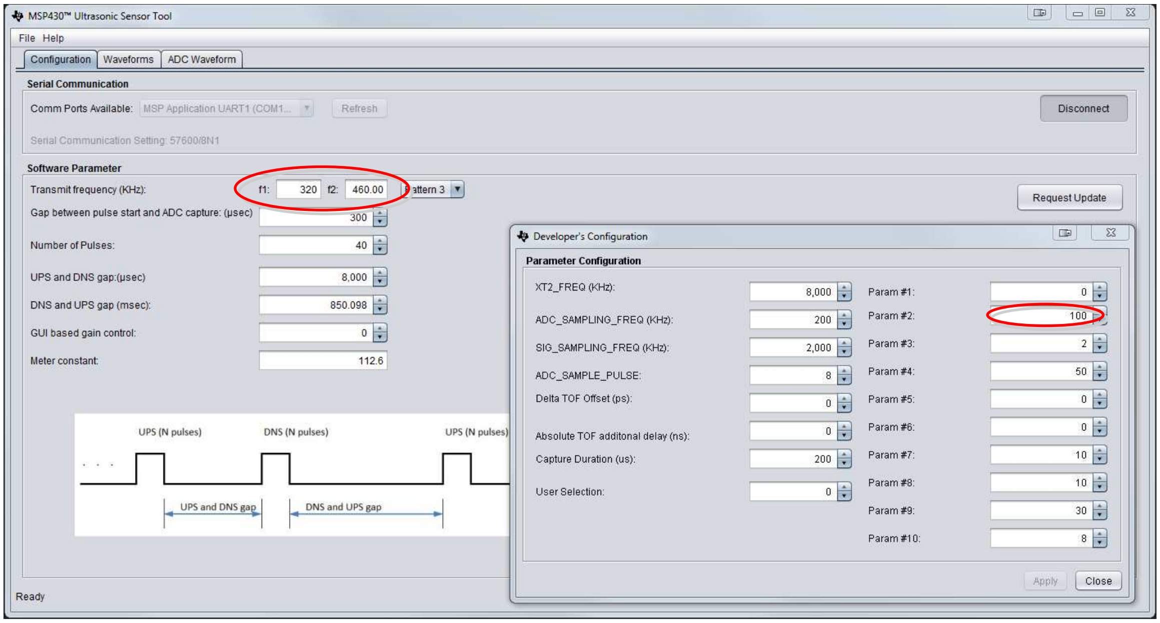


図 15. GUI Configuration for Frequency Response Test

As shown in 図 15, in order to characterize the frequency response of a meter, GUI Param #2 must be set to 100, and the *Apply* button must be selected. The range over which the frequency response characterization is to take place should then entered in the frequency boxes (320 kHz to 460kHz, in this case). The *Request Update* button should then be selected. The GUI will show that it is busy for several minutes before a popup indicating that the update was successful will appear.

At this point, there should be a file called *capturefile.csv*, which contains the frequency response waveforms in the GUI *backup* directory. This file can be imported into Matlab and processed with a simple script to provide the frequency response like:

```
d= buffer(DNS, 400);
u = buffer(UPS, 400);
f = (320:460-1);
figure(1);
plot(f, max(d) - min(d), 'b'); hold on;
plot(f, max(u) - min(u), 'r'); hold on;
```

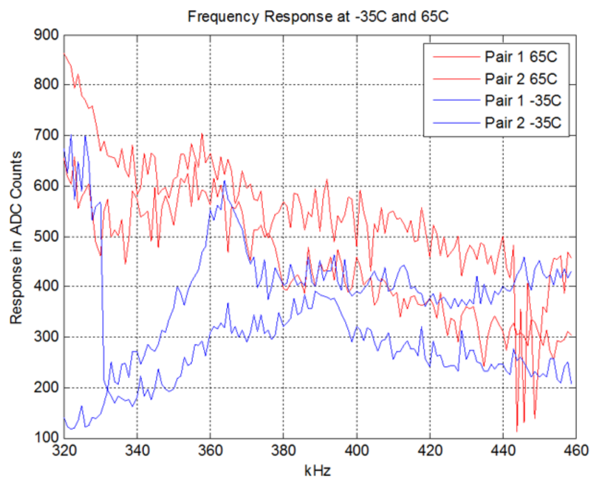


図 16. Frequency Response Over Temperature

As shown in 図 16, ultrasonic transducer frequency response shifts over temperature. Understanding how the resonance of these transducers shifts over temperature is critical to setting an appropriate frequency range for excitation, which will maximize the signal-to-noise ratio (SNR). For the CeramTec 400-kHz transducers used in this reference design, the optimal range for excitation was found to be from 340kHz to 420kHz.

When using these transducers, the GUI configuration should appear as in 図 17. Note that the software package included in this reference design includes sample configuration files located in GasLibrary folder>\examples\gui_config\.

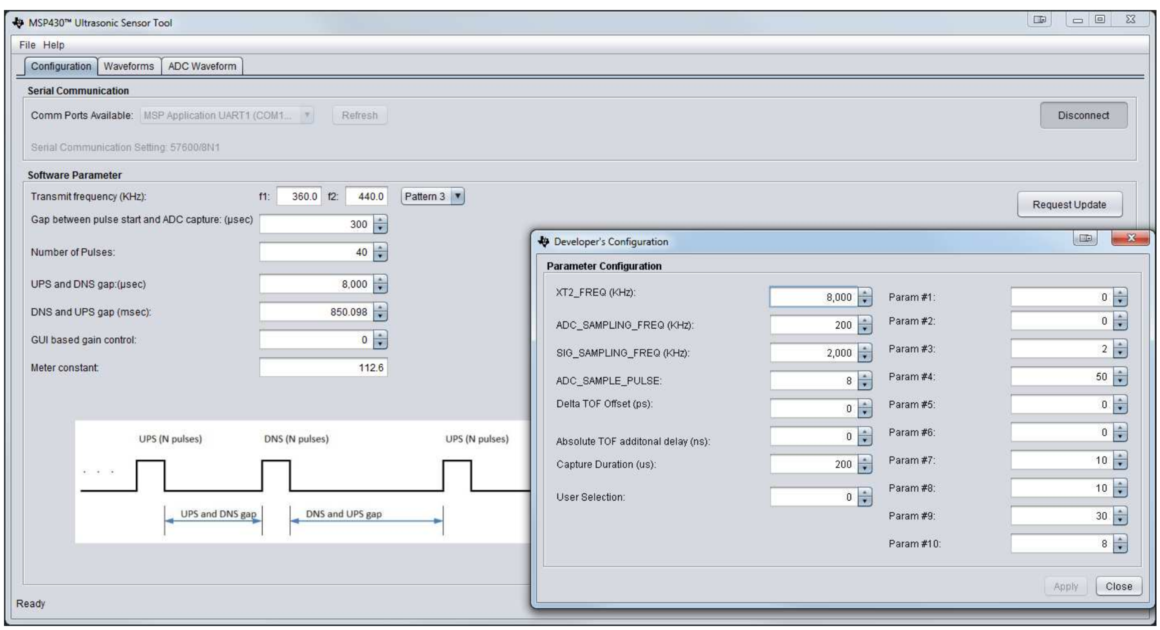


図 17. GUI Configuration Using CeramTec™ Transducers

3.2 *Testing and Results*

3.2.1 Test Setup

This section describes how to run and demo the ultrasonic gas meter front-end reference design.

3.2.1.1 Connecting the Hardware

1. Configure the ultrasonic AFE BoosterPack jumpers according to the default configuration shown in 3.1.1.2.
2. Configure the MSP-EXP430FR5994 according to the *Programming* configuration shown in 3.1.1.1.
3. Connect the ultrasonic AFE Boosterpack to the MSP-EXP430FR5994. Make sure the two 2×10 LaunchPad connectors and the two 2-pin GND connectors are aligned (as shown in 図 18).
4. Connect the transducers to J1 and J2 in the BoosterPack. If necessary, connect the meter's ground or shield to J6 or J5.
5. Connect the LaunchPad's USB to the PC.

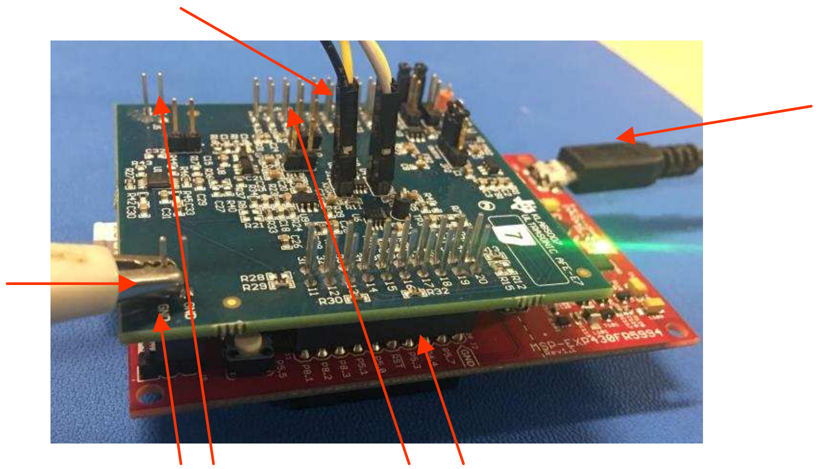


図 18. Hardware Connections

3.2.1.2 Building and Loading the Software

The software package included in this reference design includes projects for CCS and IAR. The following sections explain how to build and load the software to the device.

3.2.1.2.1 Using CCS

1. Open or create a workspace.
2. Import the application project by selecting *Menu → Project→ Import CCS Projects* and selecting the following path:<Gas Library folder>\examples\ExampleProjects\msp-expfr5994_gas_afe_bp\CCS\.

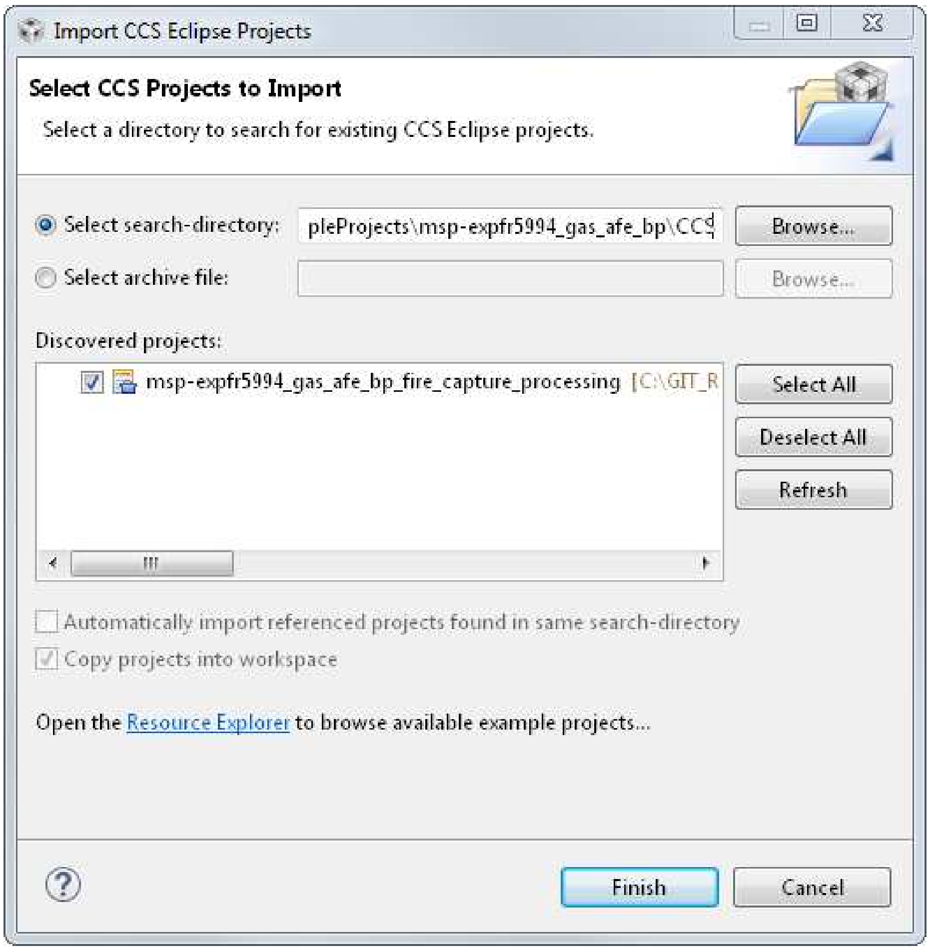


図 19. Importing Project to CCS

3. Build project (Ctrl+B).
4. Download the code to the device (F11).
5. Close the debugger. Note that it is also possible to execute or debug the application, but this design guide only shows standalone execution.

3.2.1.2.2 Using IAR

1. Open the project workspace from <Gas Library folder>\examples\ExampleProjects\msp-expfr5994_gas_afe_bp\IAR\ Gas_Metering_examples_worspace.eww.
2. Build project (F7).
3. Download the code to the device (Ctrl+D).
4. Close the debugger. Note that it is possible to execute or debug the application, but this design guide only shows standalone execution.

3.2.1.3 Execute the Application

1. To execute the application without debugger interaction, configure the LaunchPad jumpers as shown in the *Execution (USB Power)* or *Execution (Ext Power)* configurations shown in 3.1.1.1.
2. If using external power, connect a 3.3-V supply to J5 in LaunchPad.
3. Press the MSP-EXP430FR5994 reset button (S3) to reset the device and execute the application. Note that the device is programmed at this point, and the application will always execute when pressing the reset button or after a power-on-reset.

In addition to the GUI interface, which is used to configure ultrasonic sensing parameters, the example included in this reference design uses the LaunchPad's S1 to switch between a low-power mode (default) and the active mode and the two LEDs to show the status of the system as described in 表 7.

表 7. Application Execution Modes

MODE	DESCRIPTION	LED STATUS
Active	Device periodically receives new captures and calculates the required outputs successfully. The device goes to LPM0 between samples.	LED1: Blinks on every conversion (1 Hz) LED2: Off
Low-power (default after reset)	The system periodically receives new captures and calculates the required outputs successfully. The device goes to LPM3 between samples. LED1 is not toggled to minimize power consumption.	LED1: Off LED2: Off
GUI update	The GUI sends a new configuration to the device.	LED1: Off LED2: Turns on only during the duration of the GUI transfer.
Fault	An error was detected during the signal capture or algorithm processing. Only the current capture is invalid.	LED1: Blinks on every conversion (1 Hz) LED2: Turns on after the fault but will clear on the next capture.
Non-fatal error	The system detects a non-fatal error, which preventsthe system from running as expected, but the system still runs using a previous configuration. This error can happen when an incorrect configuration is received from the GUI.	LED1: Blinks on every conversion (1 Hz) LED2: On
Fatal error	The system detects a fatal error and cannot perform captures or calculate required outputs with the current configuration.	LED1: On LED2: On

3.2.1.4 Configure Device and Observe Results Using GUI

1. Open the GUI application located in <USS_GUI root>\USS_GUI.jar.
2. In the configuration window, select the COM port corresponding to the MSP-EXP430FR5994 LaunchPad.
3. Click the *Connect* button.
4. Click the *Load Configuration* button to load one of the default configurations.
5. Select one of the configurations found at <Gas Library folder>\examples\gui_config\.
6. Click the *Request Update* button to send the configuration to the device.
7. The GUI will show the message in 図 20 when the configuration is loaded to the device.

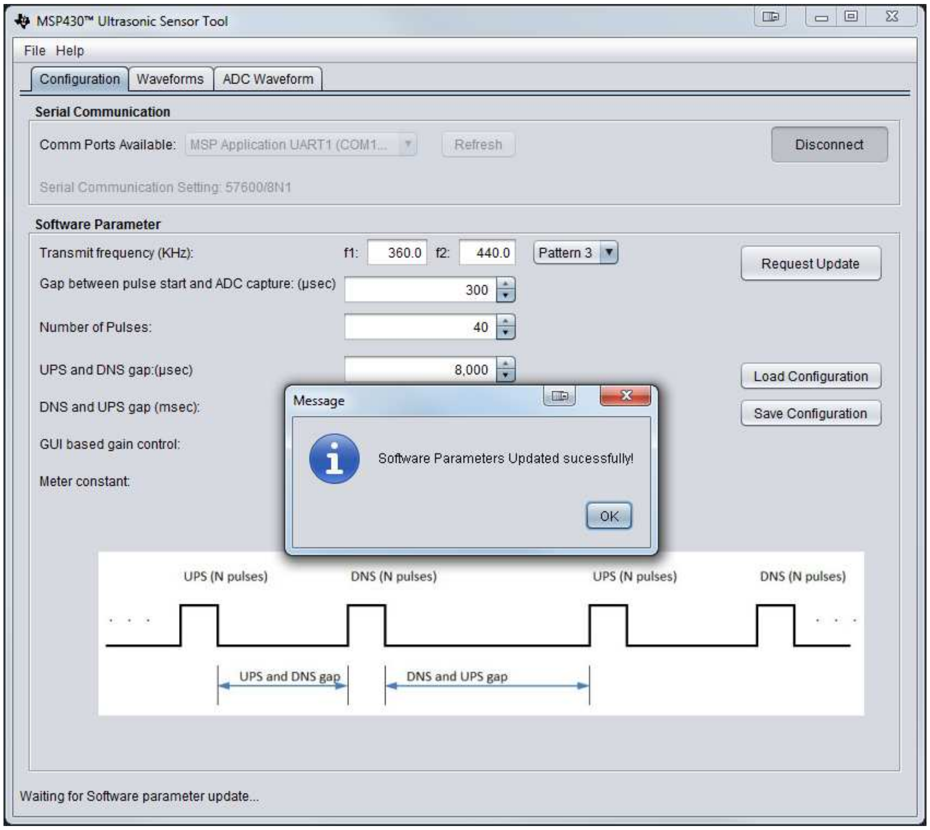


図 20. Loading a Configuration to the Device from the GUI

8. Check the integrity of the signal by using the *ADC Waveform* window as explained in 3.1.2.5.
9. Check the calculated TOF and volume in the *Waveforms* window as explained in 3.1.2.5.
10. If desired, adjust the parameters and observe the impact on the captured waveform and calculations.

3.2.1.5 Customization and Optimization

The TIDM-1002 was tested using the 3D-printed V meter described in 3.1.3; however, the system allows for an easier customization and optimization using other transducers and meter designs.

Developers are encouraged to use the MSP430 Ultrasonic Sensor Tool GUI to adjust the different configuration parameters to achieve the required balance between performance and power consumption. For example, developers can select a signal sampling frequency (SIG_SAMPLING_FREQ) of 2 MHz to achieve higher resolution at the expense of higher power consumption; however, this would be unnecessary for low-frequency transducers. In such case, developers can select a sampling frequency of 1 MHz, which results in lower power consumption.

The following GUI parameters should be adjusted based on the characteristics of the meter and transducers:

- *Transmit Frequency* depends on the excitation frequency of the transducer. For example, selecting f1 = 360 KHz and f2 = 440 KHz will generate an excitation signal modulated between these two frequencies for a 400-KHz transducer.
- *Gap between pulse start and ADC capture* depends on the dimensions of the meter and should be characterized to allow for an appropriate signal capture at different flow rates and temperature variations.
- *Number of pulses* affects the peak amplitude and, in turn, the energy received by the receiving transducer. This affects the SNR and the STD achieved by the meter; however, this parameter is also directly proportional to current consumption. Developers must adjust this value to generate an appropriate signal amplitude and standard deviation while achieving adequate current consumption.
- *UPS and DNS gap* developers must select an appropriate delay guaranteeing a clean line before the start of the next sequence.
- *GUI based gain control* selects between the two gain configurations provided by the ultrasonic AFE BoosterPack. Select 0 to enable an amplifier gain with cut-off frequency of 580 KHz (default) or 1 for a cut-off frequency of 566 KHz.
- *Meter constant* is a constant used to calculate volume flow rate as a function of the TOF and the area of a given meter as described in 式 4. A typical procedure to calculate this constant is to provide a constant flow (for example, 100 lph) and adjust the meter constant to provide the corresponding volume flow rate using the GUI.
- *ADC_SAMPLING_FREQ* is selectable between 100 KHz or 200 KHz but is typically configured to 200 KHz to achieve lower power consumption through faster signal sampling.
- *SIG_SAMPLING_FREQ* is selectable between 1 MHz or 2 MHz and must be high enough to meet Nyquist criterion for reduced error during interpolation. The recommended sampling rate is >3.5 times of the transducer frequency. Developers can modify it as a tradeoff between resolution and power consumption.
- *Delta TOF offset* adjustment is made to the differential TOF during the calculation of volume inside the ultrasonic gas library. Developers can optionally make their custom calculation of the volume flow rate based on the TOF values provided by the library.
- *Absolute TOF additional delay* adjustment made to the absolute TOF to account for any additional delays. The value depends on the shape of the signal and the envelope threshold discussed below. Developers can optionally set this value to zero and make their own adjustments of the absolute TOF in the application layer.

- *Capture duration* depends on the shape of the signal received from the transducer. Must be characterized to allow for an appropiate signal capture at different flow rates and temperature variations.
- *Param #3 (computation mode)* selects between different computation modes used by the algorithms. Refer to the Ultrasonic Sensing Gas Metering Library for more information.
- *Param #4 (envelope threshold)* adjusts the envelope threshold used by the library to detect the presence of the signal. Nominally set to 50% of the envelope but can be adjusted depending on the shape of the signal. A higher value represents a bigger delay since the actual start of the signal and this value can be subtracted by the developer or using the Absolute TOF additional delay.

3.2.2 Test Results

This section details experimental results for the 400-kHz reference meter previously described.

3.2.2.1 Single-Shot Standard Deviation

The standard deviation of the differential TOF is a measure used to express the expected variance of the output at zero-flow and ambient temperature.

表 8. Single-Shot Standard Deviation

PARAMETER	TEST CONDITIONS	TYPICAL	UNIT
Single-shot standard deviation (STD)	Using CeramTec 400-kHz transducer and 3D-printed *V* meter	2	ns
	T_A = 25ºC.		
	Zero flow		
	Capture time = 5 min ± 1 min		
	Computation mode = 2		

The STD is calculated by executing the application at ambient temperature and in a zero-flow condition. The MSP430 Ultrasonic Sensor Tool GUI can then be used to capture the differential TOF for five minutes, and the standard deviation can be calculated using a tool like Matlab or Excel.

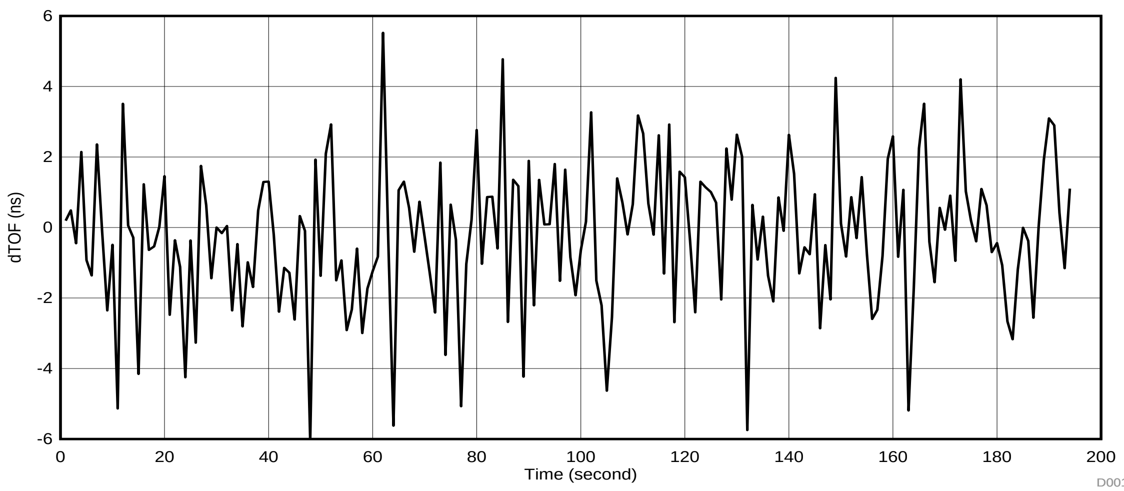


図 21. Single-Shot Standard Deviation

3.2.2.2 Zero-Flow Drift

Zero-flow drift (ZFD) is a measure used to express the expected variance of the output at zero-flow across temperature. This measure is used to determine the minimal detectable flow of the meter over temperature.

表 9. ZFD

PARAMETER	TEST CONDITIONS	TYPICAL	UNIT
ZFD	Using CeramTec 400-kHz transducer and 3D-printed *V* meter	1	ns
	T_A = –30ºC to 60ºC		
	Zero flow		
	Capture time = approximately 10 hours		
	Computation mode = 2		

ZFD tests are conducted over a –30ºC to 60ºC range. The oven profile used to calculate ZFD is observed in 図 22 together with the expected change on absolute TOF. As observed, the absolute TOF is expected to decrease with increasing temperature.

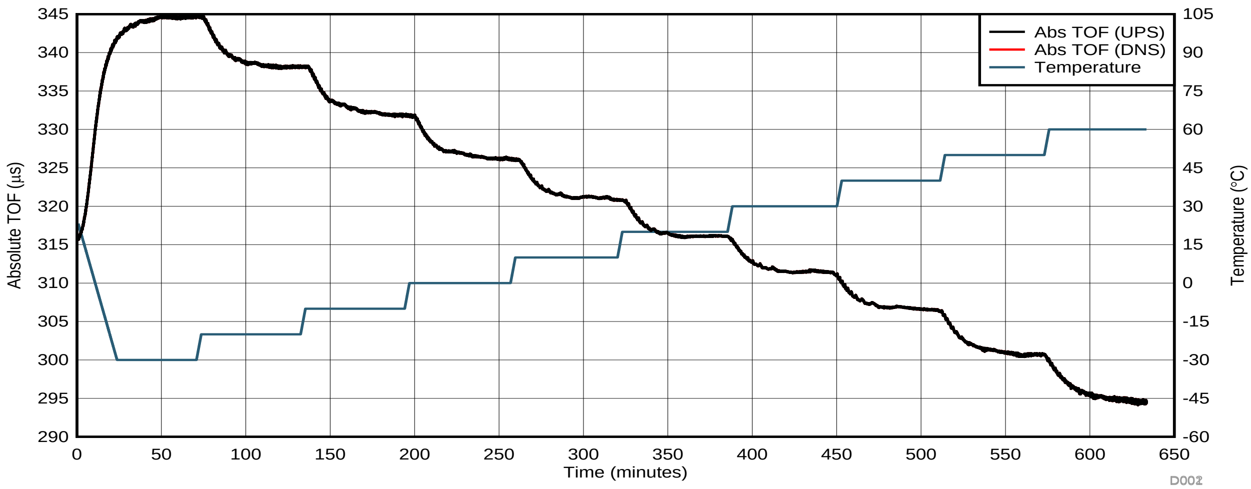


図 22. Temperature Profile for ZFD

ZFD is calculated by obtaining the range of the differential TOF averaged over 200 samples. The MSP430 Ultrasonic Sensor Tool GUI can be used to capture the differential TOF while tools like Matlab or Excel can be used to calculate the average over 200 samples and the total drift.

図 23 shows the resulting differential TOF across temperature averaged over 200 samples.

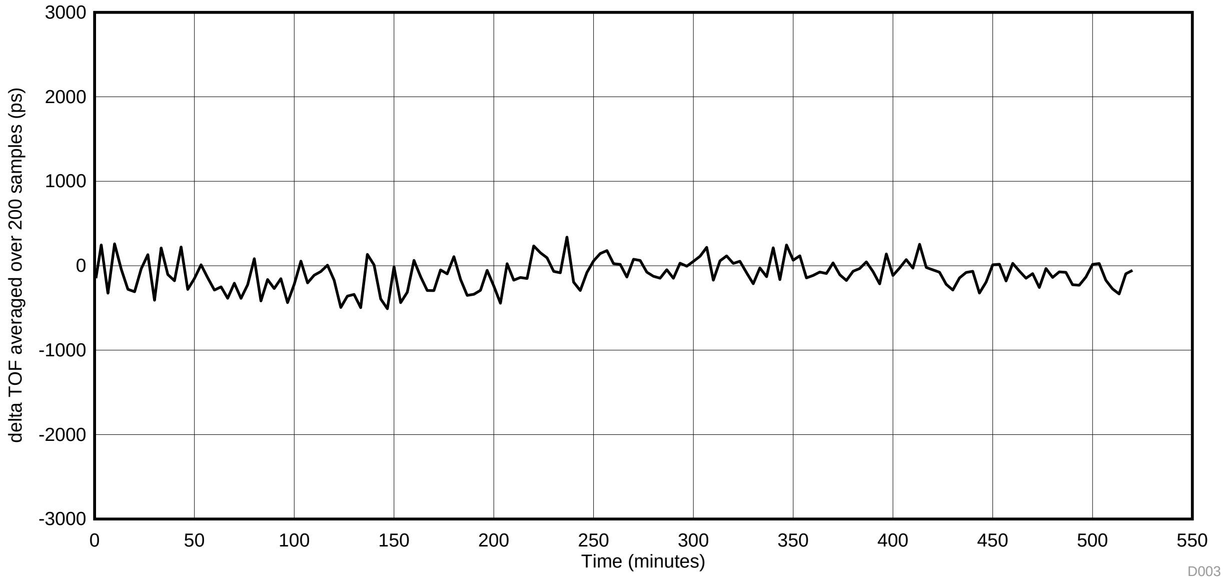


図 23. Delta TOF for ZFD

ZFD can vary depending on the meter design and the impedance matching of the transducers. 図 24 shows the ZFD for unmatched transducers. Note that the differential TOF is within 1 ns except for jumps during temperature changes.

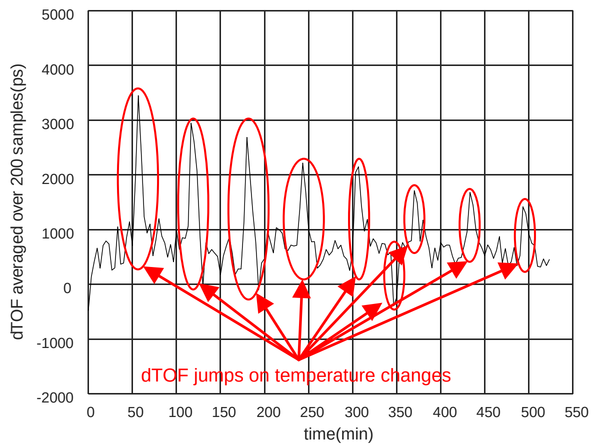


図 24. ZFD for Unmatched Transducer

This behavior can be mitigated by proper meter design and impedance matching of the transducers.

3.2.2.3 Minimum Detectable Flow

The minimum detectable flow (MDF) uses the ZFD results to determine the minimal detectable flow of the meter over temperature.

表 10. MDF

PARAMETER	TEST CONDITIONS	TYPICAL	UNIT
MDF	Using CeramTec 400-kHz transducer and 3D-printed *V* meter	7.95	lph
	T_A = -30ºC to 60ºC		
	MaxFlow = 2.9 cfm (4930 lph)		
	ZFD = 1 ns		
	$dTOF_{maxflow}$= 6.2 µs		
	Error = 10%		
	Computation mode = 2		

Assuming an error of 10%, the MDFcan be calculated with 式 5.

$$MDF = \frac{MaxFlow \times (ZFD \times Error)}{dTOF_{max\,flow}} \tag{5}$$

Assuming an error of 10%, a ZFD of 1ns (as shown in 3.2.2.2), and a differential TOF of 6.2µs at the maximum flow of 4930lph, the minimum detectable flow can be calculated by 式 6.

$$\frac{4390 \times (1\,ns \times 10)}{6.2\ \mu s} = 7.95\ lph \tag{6}$$

Flow measurements were performed using a fan pre-calibrated using a reference meter. The results of the differential TOF at different flow rates are observed in 図 25 and 図 26.

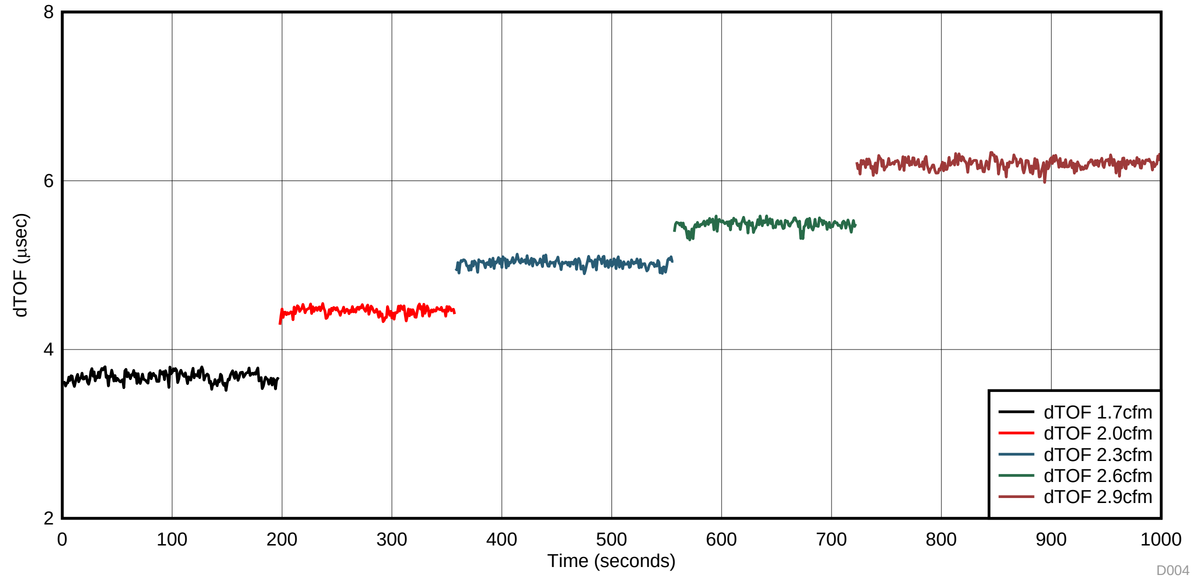


図 25. dTOF at Different Flow Rates

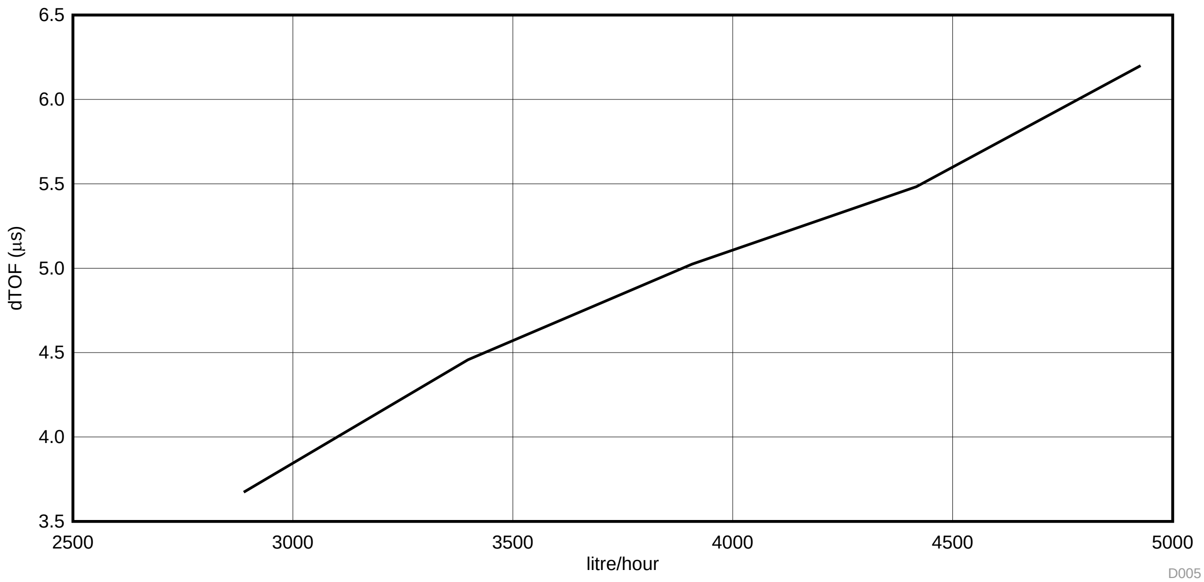


図 26. dTOF at Different Flow Rates

A more linear curve can be obtained based on the meter design.

3.2.2.4 Average Power Consumption

The average power consumption is an important consideration for power-restricted systems, such as battery-operated meters.

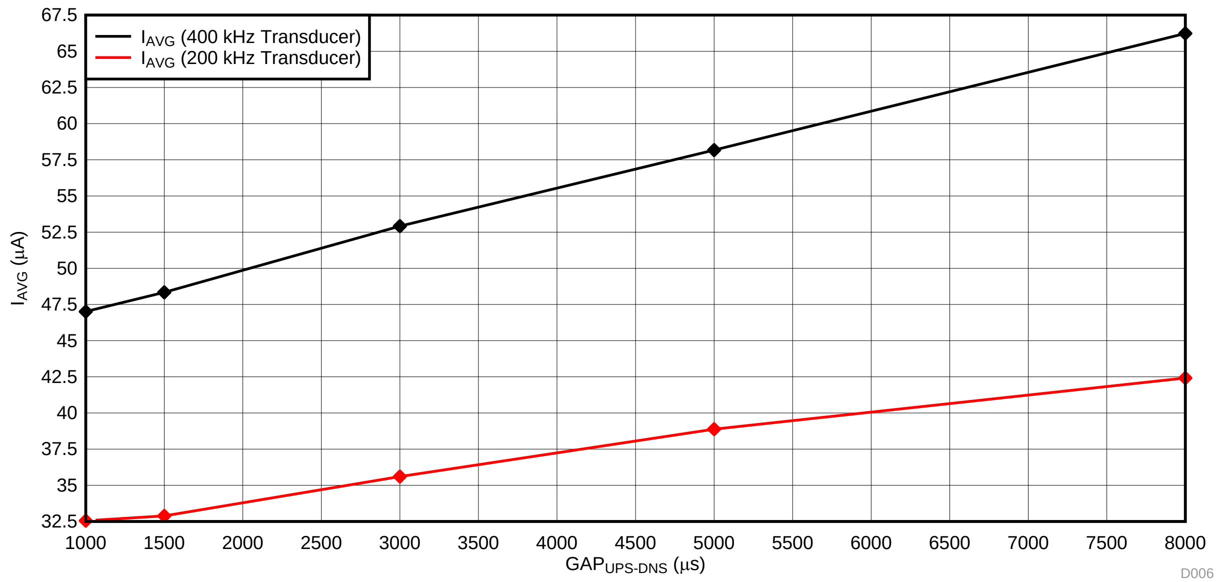
表 11. Average Power Consumption

PARAMETER	TEST CONDITIONS	TYPICAL	UNIT
Average current consumption (I_{AVG})	Using CeramTec 400-kHz transducer and 3D-printed *V* meter	48.3	µA
	T_A = 25ºC		
	Zero flow		
	Measurement rate: 1 per second		
	Low-power mode		
	Gap between UPS and DNS = 1.5 ms		
	Computation mode = 2		

The power consumption of the device will depend on several different parameters, which can be configured using the MSP430 Ultrasonic Sensor Tool GUI. These parameters include transducer frequency, number of pulses, gap between UPS and DNS, measurement rate (DNS-UPS gap), signal sampling frequency, capture duration, and so forth.

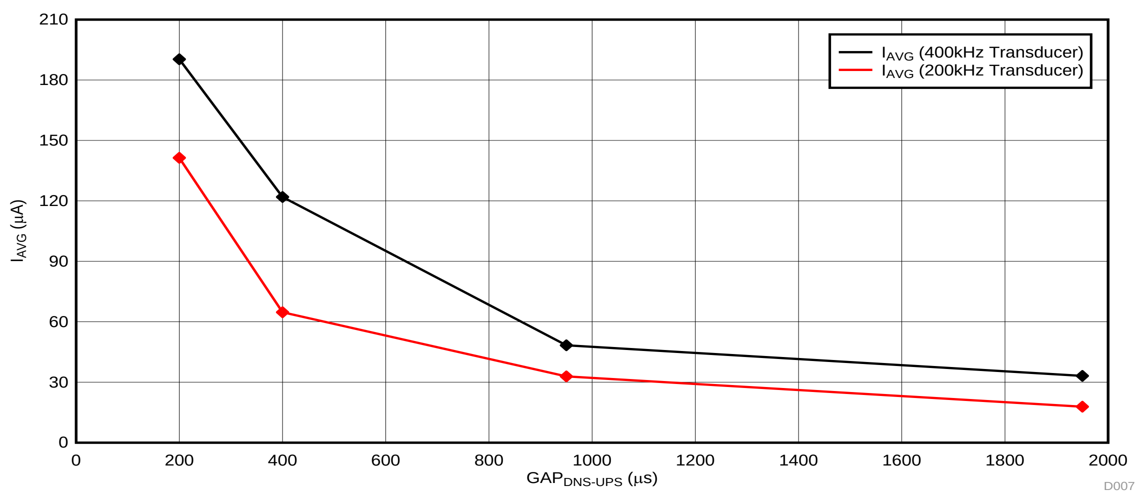
The optimal configuration of these parameters will depend on the meter itself and a tradeoff between performance and power consumption.

図 27 and 図 28 show the average current consumption of the system under with different configurations.



Test Conditions:
400kHz Transducer: Number of pulses = 40, $GAP_{UPS-DNS}$ = 950, Signal sampling frequency = 2 Msps, Capture duration = 200 µs
200kHz Transducer: Number of pulses = 20, $GAP_{UPS-DNS}$ = 950, Signal sampling frequency = 1 Msps, Capture duration = 400 µs

図 27. Current Consumption Versus $GAP_{UPS-DNS}$



Test Conditions:
400kHz Transducer: Number of pulses = 40, $GAP_{DNS-UPS}$ = 1500, Signal sampling frequency = 2 Msps, Capture duration = 200 µs
200kHz Transducer: Number of pulses = 20, $GAP_{DNS-UPS}$ = 1500, Signal sampling frequency = 1 Msps, Capture duration = 400 µs

図 28. Current Consumption Versus $GAP_{DNS-UPS}$

3.2.2.5 Memory Footprint

The memory footprint provides an estimate of the memory requirements to implement an ultrasonic meter application.

表 12. Memory Footprint

PARAMETER	TEST CONDITIONS	TYPICAL		UNIT
		IAR[(1)]	CCS[(2)]	
Total NVM used by demonstration application (MEM_{APP_NVM})	Includes code and constants	57.1	69.2	KB
Total RAM used by demonstration application (MEM_{APP_RAM})	Includes variables placed in RAM and FRAM	5.3	5.3	KB
NVM used by library (MEM_{LIB_NVM})	Includes all library code and constants used by application	25.8	21.9	KB
RAM used by library (MEM_{LIB_RAM})	Includes all library variables used by application	4.4	4.4	KB

(1) Using IAR 6.40.4 with medium optimization level
(2) Using CCS 6.2.0 with optimization level 3

The application demonstration discussed in this reference design includes some functionality, which is not required for a final application (such as GUI communication).

The library estimates shown in 表 12 include all modules residing in the Ultrasonic Sensing Gas Metering Library and the gasMeteringHAL and gasMeteringUserConfig modules, which are located in the application as source code.

4 Design Files

4.1 Schematics

To download the schematics, see the design files at TIDM-1002.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDM-1002.

4.3 PCB Layout Prints

To download the layout prints for each board, see the design files at TIDM-1002.

4.4 Altium Project

To download the Altium project files, see the design files at TIDM-1002.

4.5 Gerber Files

To download the Gerber files, see the design files at TIDM-1002.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDM-1002.

5 Software Files

To download the software files, see the design files at TIDM-1002.

6 Related Documentation

1. Texas Instruments, *Low-Energy Accelerator (LEA) Frequently Asked Questions (FAQ)* , Application Note (SLAA720)
2. Texas Instruments, *Benchmarking the Signal Processing Capabilities of the Low-Energy Accelerator on MSP430 MCUs* , Application Report (SLAA698)

6.1 商標

7 About the Authors

LUIS REYNOSO is a systems applications engineer on the MSP Smart Grid Systems team at Texas Instruments. He has taken multiple roles in the embedded industry and during this time he has published several applications notes and papers for MCUs. He joined the MSP430 Applications team in 2010.

LEO ESTEVEZ is a systems applications engineer on the MSP Smart Grid Systems team at Texas Instruments. He has developed a variety of embedded systems. During his time at TI he has conducted research in computer vision, wireless, ultrasound, and bio-metric sensing. He joined Texas Instruments DSP R&D labs in 1997.

DOMINGO GARCIA is a systems engineer in TI's Kilby Labs. He enjoys designing the initial prototypes that implement new ideas and algorithms. He joined Kilby Labs in 2015.

ANAND DABAK is systems engineer in TI's Kilby labs. He is a TI Fellow and joined Texas Instruments DSP R&D labs in 1995. During his time at TI he has worked on wireless communications, powerline communications, and ultrasonic flow metering.

TIの設計情報およびリソースに関する重要な注意事項

Texas Instruments Incorporated ("TI")の技術、アプリケーションその他設計に関する助言、サービスまたは情報は、TI製品を組み込んだアプリケーションを開発する設計者に役立つことを目的として提供するものです。これにはリファレンス設計や、評価モジュールに関係する資料が含まれますが、これらに限られません。以下、これらを総称して「TIリソース」と呼びます。いかなる方法であっても、TIリソースのいずれかをダウンロード、アクセス、または使用した場合、お客様(個人、または会社を代表している場合にはお客様の会社)は、これらのリソースをここに記載された目的にのみ使用し、この注意事項の条項に従うことに合意したものとします。

TIによるTIリソースの提供は、TI製品に対する該当の発行済み保証事項または免責事項を拡張またはいかなる形でも変更するものではなく、これらのTIリソースを提供することによって、TIにはいかなる追加義務も責任も発生しないものとします。TIは、自社のTIリソースに訂正、拡張、改良、およびその他の変更を加える権利を留保します。

お客様は、自らのアプリケーションの設計において、ご自身が独自に分析、評価、判断を行う責任がお客様にあり、お客様のアプリケーション(および、お客様のアプリケーションに使用されるすべてのTI製品)の安全性、および該当するすべての規制、法、その他適用される要件への遵守を保証するすべての責任をお客様のみが負うことを理解し、合意するものとします。お客様は、自身のアプリケーションに関して、(1) 故障による危険な結果を予測し、(2) 障害とその結果を監視し、および、(3) 損害を引き起こす障害の可能性を減らし、適切な対策を行う目的での、安全策を開発し実装するために必要な、すべての技術を保持していることを表明するものとします。お客様は、TI製品を含むアプリケーションを使用または配布する前に、それらのアプリケーション、およびアプリケーションに使用されているTI製品の機能性を完全にテストすることに合意するものとします。TIは、特定のTIリソース用に発行されたドキュメントで明示的に記載されているもの以外のテストを実行していません。

お客様は、個別のTIリソースにつき、当該TIリソースに記載されているTI製品を含むアプリケーションの開発に関連する目的でのみ、使用、コピー、変更することが許可されています。明示的または黙示的を問わず、禁反言の法理その他どのような理由でも、他のTIの知的所有権に対するその他のライセンスは付与されません。また、TIまたは他のいかなる第三者のテクノロジまたは知的所有権についても、いかなるライセンスも付与されるものではありません。付与されないものには、TI製品またはサービスが使用される組み合わせ、機械、プロセスに関連する特許権、著作権、回路配置利用権、その他の知的所有権が含まれますが、これらに限られません。第三者の製品やサービスに関する、またはそれらを参照する情報は、そのような製品またはサービスを利用するライセンスを構成するものではなく、それらに対する保証または推奨を意味するものでもありません。TIリソースを使用するため、第三者の特許または他の知的所有権に基づく第三者からのライセンス、あるいはTIの特許または他の知的所有権に基づくTIからのライセンスが必要な場合があります。

TIのリソースは、それに含まれるあらゆる欠陥も含めて、「現状のまま」提供されます。TIは、TIリソースまたはその仕様に関して、明示的か暗黙的かにかかわらず、他のいかなる保証または表明も行いません。これには、正確性または完全性、権原、続発性の障害に関する保証、および商品性、特定目的への適合性、第三者の知的所有権の非侵害に対する黙示の保証が含まれますが、これらに限られません。

TIは、いかなる苦情に対しても、お客様への弁護または補償を行う義務はなく、行わないものとします。これには、任意の製品の組み合わせに関連する、またはそれらに基づく侵害の請求も含まれますが、これらに限られず、またその事実についてTIリソースまたは他の場所に記載されているか否かを問わないものとします。いかなる場合も、TIリソースまたはその使用に関連して、またはそれらにより発生した、実際的、直接的、特別、付随的、間接的、懲罰的、偶発的、または、結果的な損害について、そのような損害の可能性についてTIが知らされていたかどうかにかかわらず、TIは責任を負わないものとします。

お客様は、この注意事項の条件および条項に従わなかったために発生した、いかなる損害、コスト、損失、責任からも、TIおよびその代表者を完全に免責するものとします。

この注意事項はTIリソースに適用されます。特定の種類の資料、TI製品、およびサービスの使用および購入については、追加条項が適用されます。これには、半導体製品(http://www.ti.com/sc/docs/stdterms.htm)、評価モジュール、およびサンプル(http://www.ti.com/sc/docs/sampterms.htm)についてのTIの標準条項が含まれますが、これらに限られません。