




# **MSP430 Gas Metering library version 1.10.00.11**

## **USER'S GUIDE**

---

# Copyright

Copyright © Texas Instruments Incorporated. All rights reserved.

 Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this document.

Texas Instruments  
Post Office Box 655303  
Dallas, TX 75265  
<http://www.ti.com/msp430>



## Revision Information

This is version 1.10.00.11 of this document, last updated on April 26, 2017.

## Document License

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 United States License (CC BY-SA 3.0). To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/us/> or send a letter to

Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

## Contributors to this document

Copyright © 2015 Texas Instruments Incorporated - <http://www.ti.com/>

# Table of Contents

<b>Copyright</b>	<b>2</b>
<b>Revision Information</b>	<b>2</b>
<b>Document License</b>	<b>2</b>
<b>Contributors to this document</b>	<b>2</b>
<b>1 Introduction</b>	<b>5</b>
1.1 SW architecture and Library Usage Model	5
1.2 Firing and Capture Timing	7
1.2.1 "gapUPSandDNS_usec" parameter	8
1.2.2 "gapPlsSrtandADCCap_usec" parameter	9
1.2.3 "captureDuration_usec" parameter	10
1.2.4 "gapDNSandUPS_msec" parameter	11
1.3 msp-expfr5994_gas_afe_bp example project	12
1.3.1 Example project assumptions	12
1.3.2 Example project directory/file description	12
1.3.3 Gas Metering Library Example Project Resource Usage	14
1.4 Supported IDEs	16
<b>2 Configuration API</b>	<b>17</b>
2.0.1 Configuration_api	17
<b>3 Measurement API</b>	<b>19</b>
3.0.2 Measurement_api	19
<b>4 Algorithms API</b>	<b>21</b>
4.0.3 Algorithm_api	21
<b>IMPORTANT NOTICE</b>	<b>24</b>

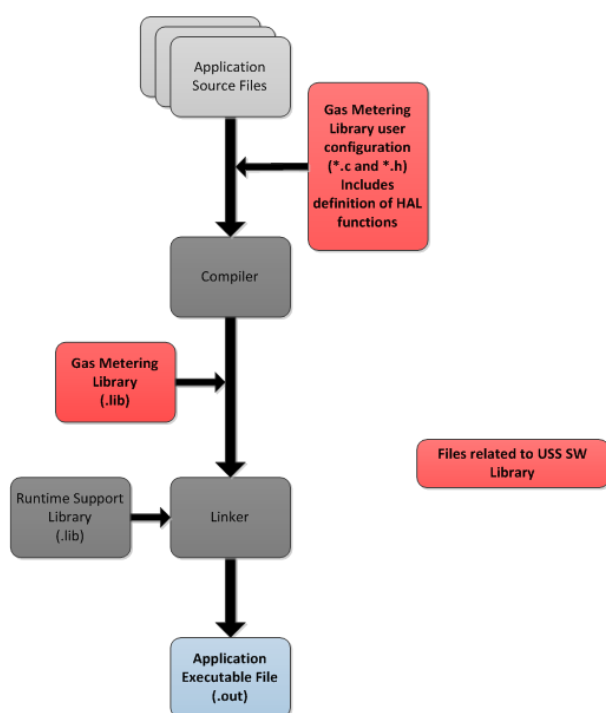


# 1 Introduction

Library provides the means to configure the internal ADC for ultrasonic captures and helps users to integrate it with the application software for development of Gas Meter Ultrasonic Flow Meter applications.

The library also takes advantage of LEA module in MSP43FR5994 to compute Time of Flight (ToF) measurements.

## 1.1 SW architecture and Library Usage Model



The Gas Metering Library consist of two parts, precompiled code and library configuration files.

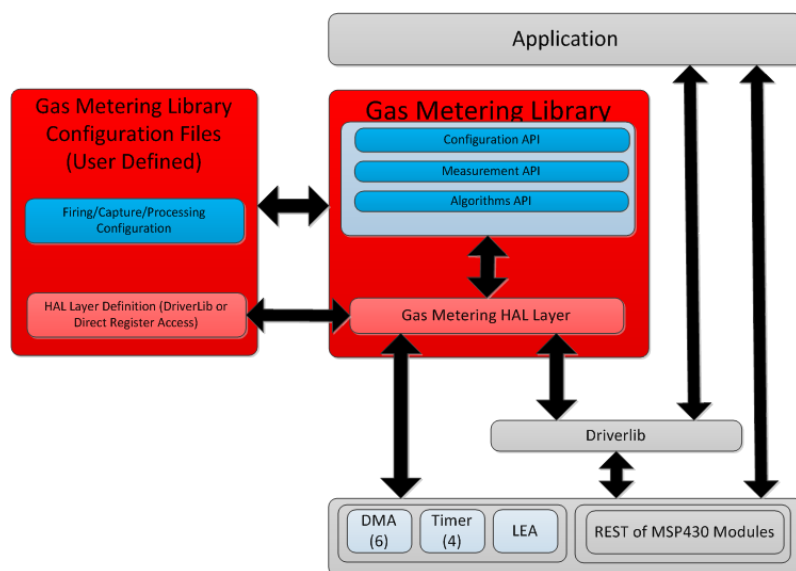
- Gas Metering Library: (Precompiled code)

- Contains:

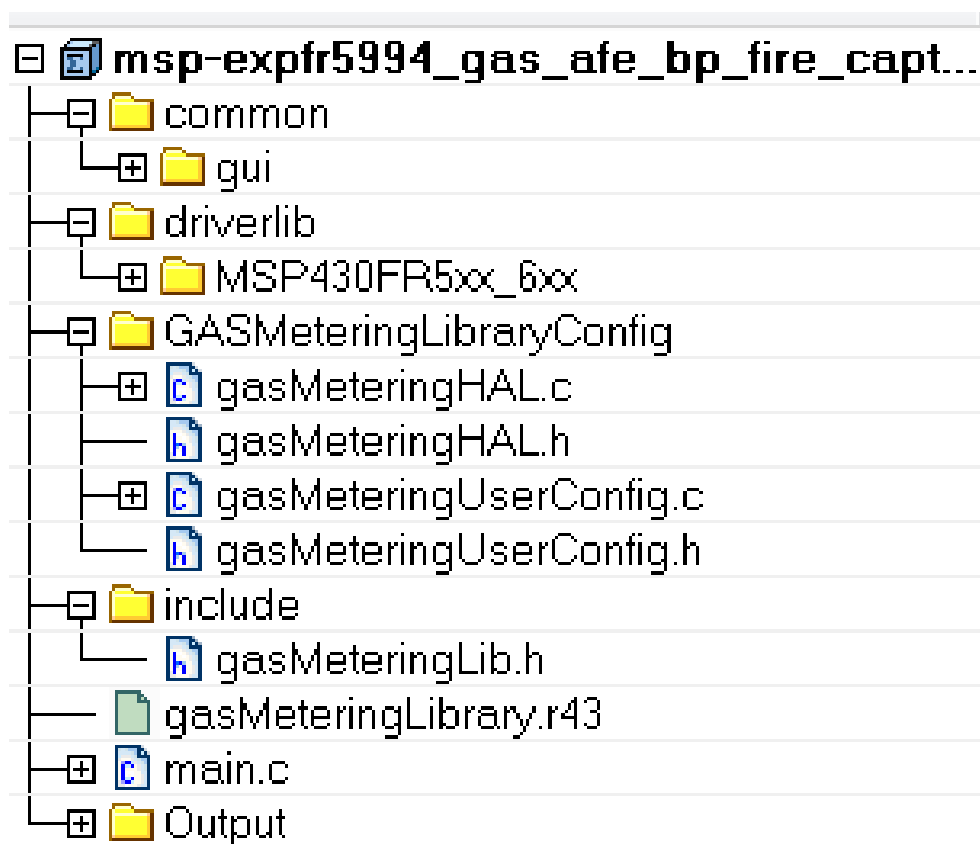
- \* Code to perform configuration/firing/capture of ultrasonic signal using interleaving technique
    - \* Code to run Absolute ToF, Delta ToF and Volume calculation

- Gas Metering Library configuration files

- All library configuration parameters are defined in these two files
  - HAL layer is also planned to be part of the files

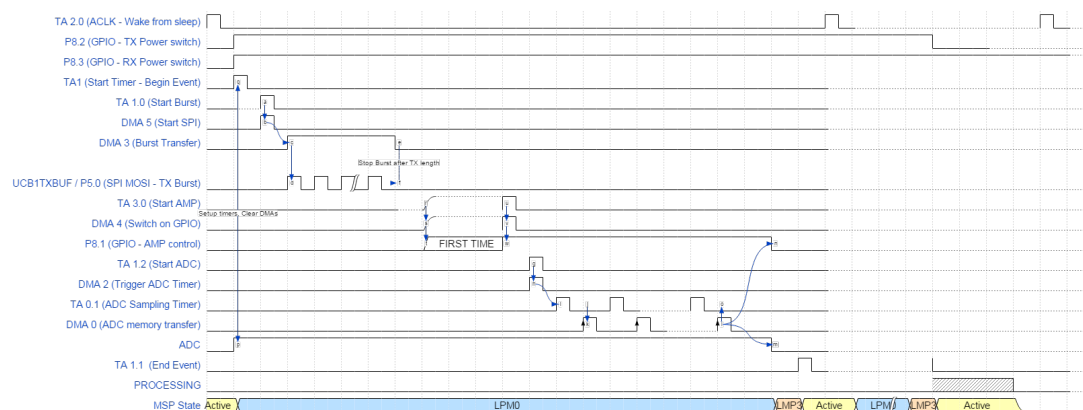


The image below shows the directory structure of the example project which is provided in the library. For more details please refer to [Example project directory/file description](#)

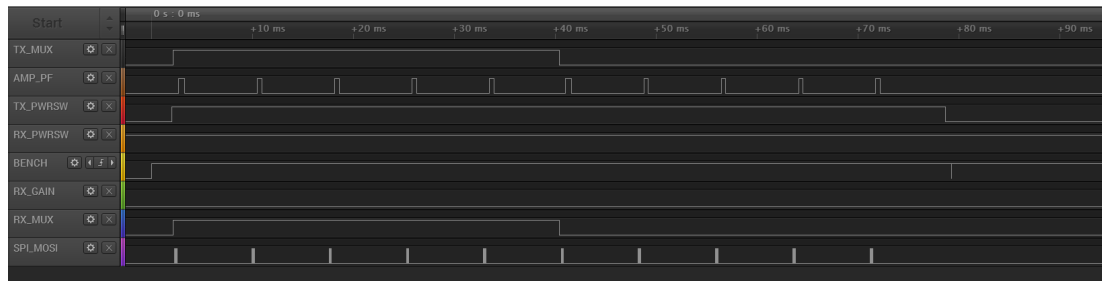


## 1.2 Firing and Capture Timing

The following diagram shows how the firing and capture are controlled by the various MSP modules used by the library.



The next timing diagram shows the complete firing/capture sequence.



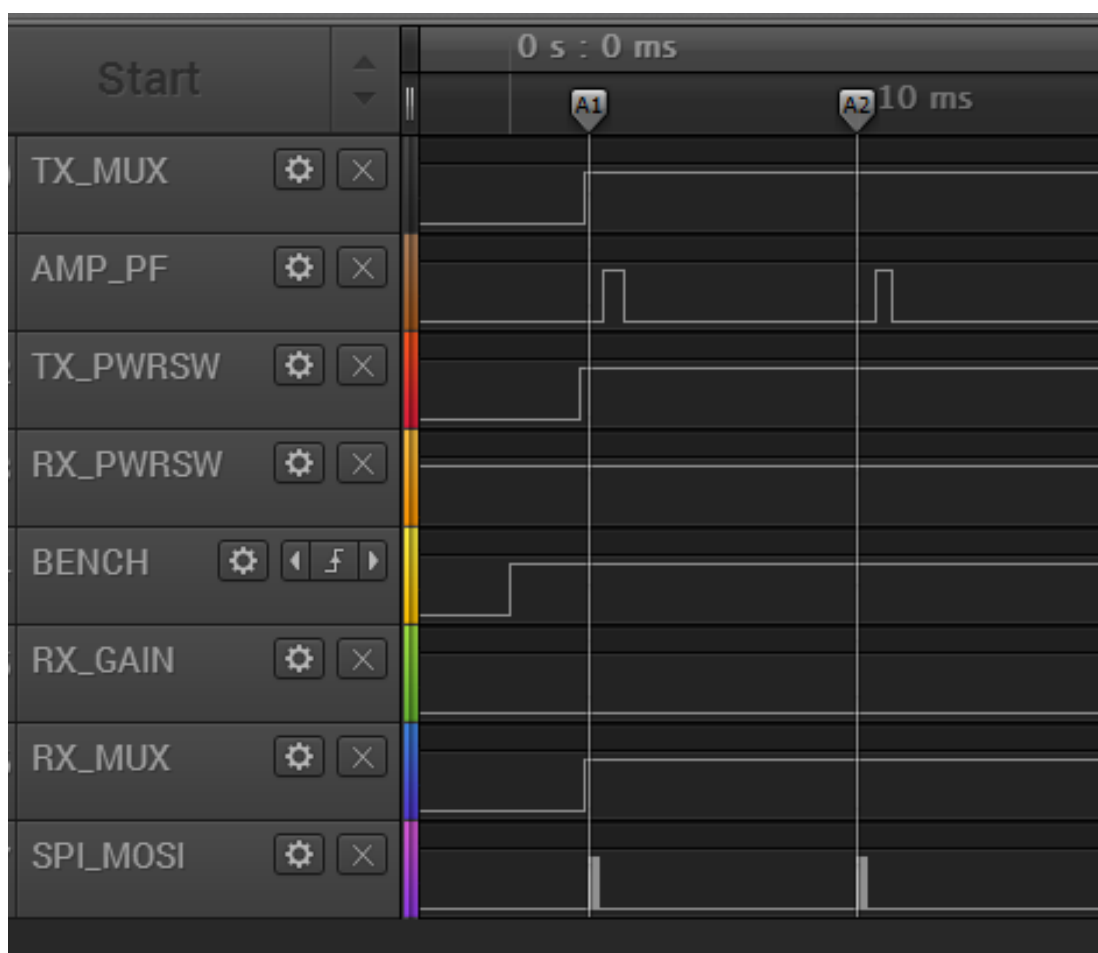
The firing and capture sequence are mainly controlled by the following configuration parameters:

- gapUPSandDNS\_usec
- gapPlsSrtandADCCap\_usec
- captureDuration\_usec
- gapDNSandUPS\_msec

The following timing diagrams shows the which part of the firing/capture sequence does each parameter controls.

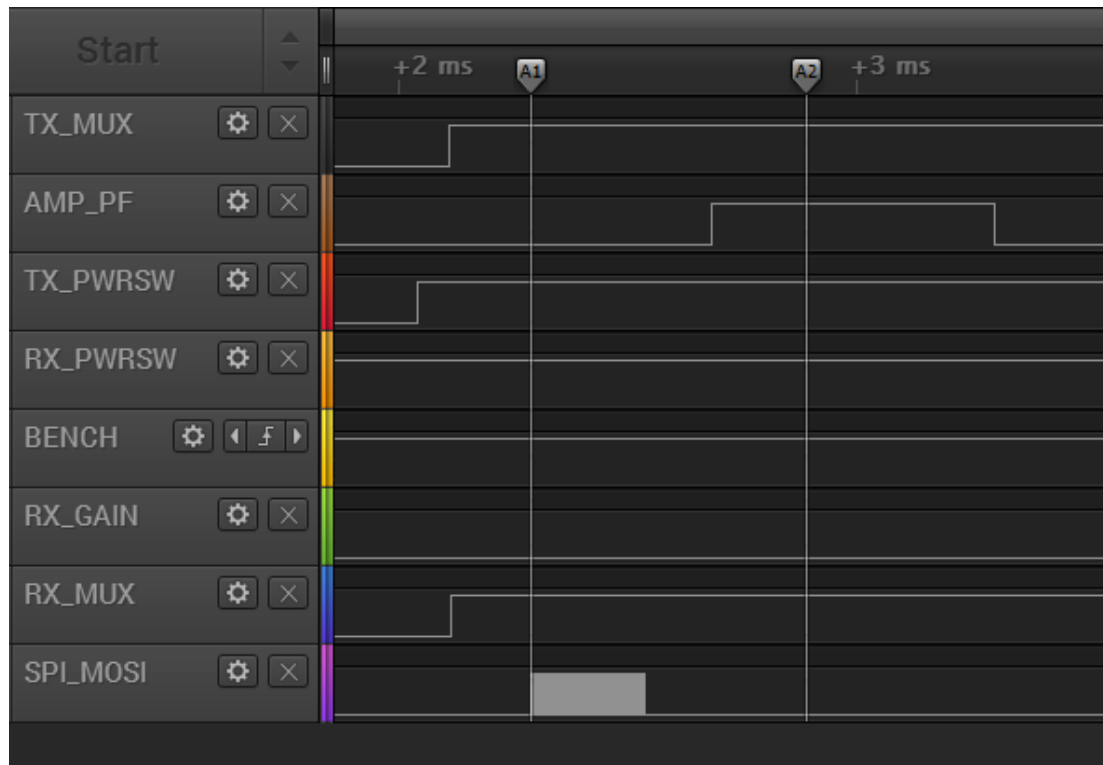
### 1.2.1 "gapUPSandDNS\_usec" parameter

The following parameter controls the gaps between a Up sample firing and a Down sample firing. In addition the same gap is used between Up sample and Down sample iteration during ADC interleaving. The image below shows the this gap.



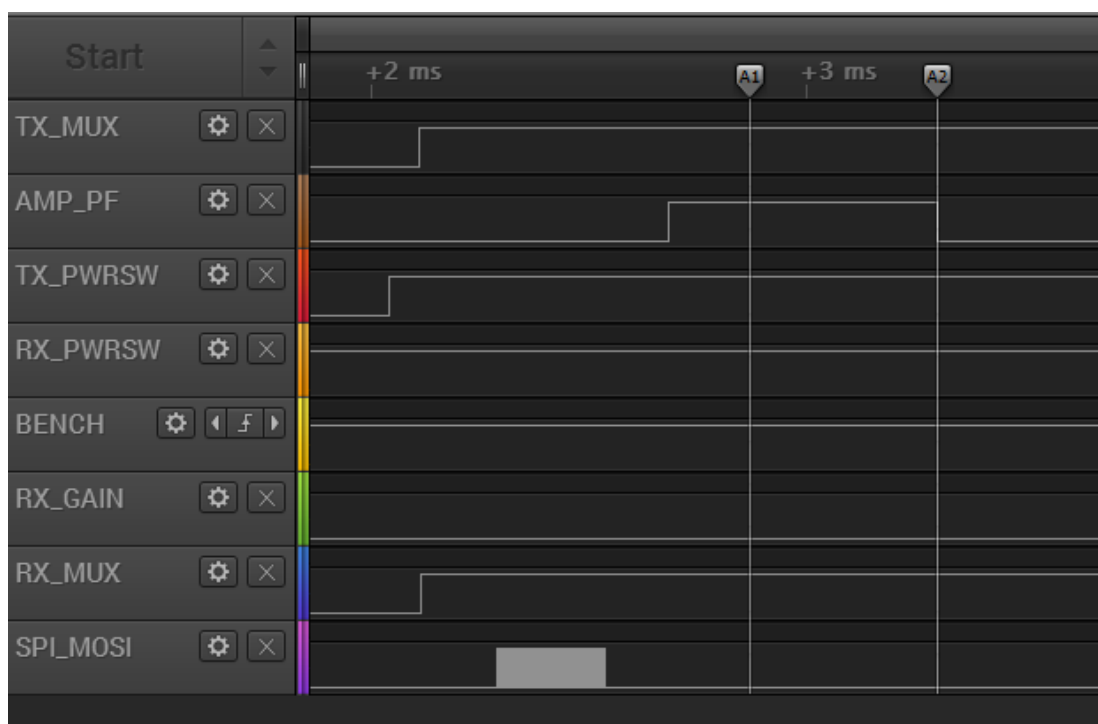
### 1.2.2 "gapPlsSrtandADCCap\_usec" parameter

This parameter control the gap between transmitting the transducer excitation pulses and triggering the ADC capture. The image below shows this gap.



### 1.2.3 "captureDuration\_usec" parameter

This parameter controls the ADC capture duration. This gap is currently timed by using the Amplifier gain line as a reference. The amplifier gain line is asserted 200 us before triggering the first ADC capture and 30 us for the rest of the ADC captures. The image below shows the ADC capture duration during the first ADC capture.



#### 1.2.4 "gapDNSandUPS\_msec" parameter

This parameter controls the gap to trigger the next firing sequence. The delay is relative to the end ToF computation. The image below shows this gap.



## 1.3 msp-expfr5994\_gas\_afe\_bp example project

### 1.3.1 Example project assumptions

**IMPORTANT** It is assumed that the example project will be used in combination with

- USS\_GUI version 1\_00\_02\_04.

Failure to use USS\_GUI will result in erroneous transducer excitation frequency generation and invalid Time of Flight computation.

#### 1.3.1.1 Using example project in IAR

Open IAR and follow these steps:

- Go to File->Open->Workspace...
- Go to the following directory <GAS\_METERING\_INSTALL\_PATH>/examples/ExampleProjects/msp-expfr5994\_gas\_afe\_bp/IAR
- Select "Gas\_Metering\_examples\_worspace.eww"

#### 1.3.1.2 Using example project in CCS

Open CCS and follow these steps:

- Go to File->Import...
- Expand Code Composer Studio
- Select CCS Project and click "Next"
- Make sure "Select search directory" is selected and click "Browse"
- Go to the following directory <GAS\_METERING\_INSTALL\_PATH>/examples/ExampleProjects/msp-expfr5994\_gas\_afe\_bp/CCS and click "OK"
- In the "Discovered project window", confirm msp-expfr5994\_gas\_afe\_bp\_fire\_capture\_processing project is selected and click "Finish"

### 1.3.2 Example project directory/file description

IAR and CCS project have the same directory/file structure:

- common
- driverlib
- GASMeteringLibraryConfig
- include
- main.c

### 1.3.2.1 "common" Directory description

This directory contains all source code to enable communication with USS\_GUI.

### 1.3.2.2 "driverlib" Directory description

This directory contains all MSP430FR5xx\_6xx Driver Library's API source files. Gas Metering Library takes advantage of MSP430 Driver Library to define Gas Metering HAL layer. MSP Driver Library keeps you above the bits and bytes of the MSP430 hardware which increases portability of your code. For more information regarding MSP Driver Library refer to driver lib product page MSP Driver Library

### 1.3.2.3 "GASMeteringLibraryConfig" Directory description

These directory contains two type of files:

- gasMeteringHAL
- gasMeteringUserConfig

#### 1.3.2.3.1 "gasMeteringHAL" files description

gasMeteringHAL.c and gasMeteringHAL.h allows the user to modify GPIO, ADC, DMA, Timers, SPI peripheral configuration via these files. The HAL layer files have been predefine to work with the MSP-EXP430FR5994 hardware. If the example project will be run on a custom board these files needs to be updated if the custom board pin configuration does not match the launchpad pin configuration.

#### 1.3.2.3.2 "gasMeteringUserConfig" files description

gasMeteringUserConfig.c and gasMeteringUserConfig.h files allows user to modify Gas Metering Library configurations. All firing, capture, meter and algorithm configuration are contained in these files. For more information regarding each configuration parameter please refer to the Gas Metering Library API User's Guide which can be found in the following directory: <GAS\_METERING\_INSTALL\_PATH>/gasLibrary-ApiGuide.html

### 1.3.2.4 "main.c" file description

main.c provides an application skeleton code and recommended Gas Metering API usage in addition to some benchmarking code which allows to easily benchmark ultrasonic measurement and algorithm computation by monitoring GPIO P1.0 which is accessible via MSP-EXP430FR5994 J7 jumper.

### 1.3.3 Gas Metering Library Example Project Resource Usage

#### 1.3.3.1 Pin I/O Usage

The total number of pins used are 9. The table below lists the details of I/O used by the SW on the MSP-EXP430FR5994 LaunchPad

MSP-EXP430FR5994 I/O Usage			
Pins	Direction I = Input O = Output	Schematic	Comments
P5.0	O	MOSI (SPI_TX)	
P8.1	O	AMP_PD	
P8.2	O	TX_PWRSW	
P8.3	O	RX_PWRSW	
P5.7	O	TX_MUX	
P4.4	O	RX_MUX	
P6.2	O	5VUP_ENA	This GPIO is reserved for future implementation
P6.3	O	RX_GAIN	
P1.3	I	PGA (2nd stage)	
P2.0	O	UART_TX	GUI_UART Communication
P2.1	I	UART_RX	GUI_UART Communication

#### 1.3.3.2 Peripheral Usage

The table below lists the MSP peripherals used by the Gas Metering Library example project.

MSP-EXP430FR5994 Peripheral Usage		
Peripheral	Number	Enumerated
eUSCI	2	eUSCI_B1 (SPI_TX) and eUSCI_A0 (GUI_COM)
Timer Channels	4	TA0, TA1, TA2, TA3
DMAs	5	DMA0, DMA2, DMA3, DMA4, DMA5
ADC Channels	1	ADC12_B (Input Channel 3)

#### 1.3.3.3 Configuring measurement sample rate

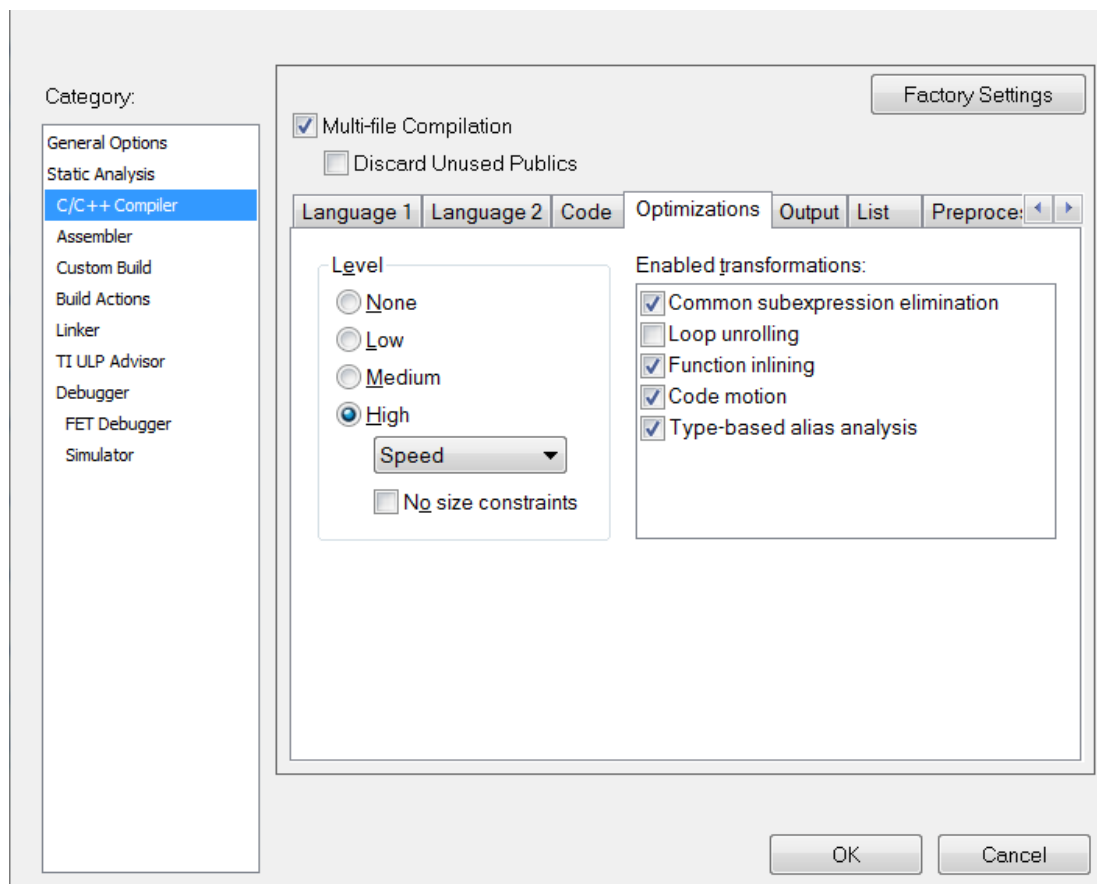
The table below provides the DNS to UPS gap configuration needed to generate 1Hz and 8Hz sample rates.

Measurement sample rate configuration								
Transmit frequency	Pattern Option	ADC sampling frequency (KSPS)	Signal Capture Frequency (KHz)	Capture Duration (usec)	Computation Mode	Desired Sample Rate (Hz)	GUI enabled	DNS to UPS gap (msec)
120KHz - 200KHz	3	200	1000	400	1	1	YES	900
120KHz - 200KHz	3	200	1000	400	1	8	YES	28
120KHz - 200KHz	3	200	1000	400	1	1	NO	907
120KHz - 200KHz	3	200	1000	400	1	8	NO	31

### 1.3.3.4 Example Project Memory Usage Details

#### 1.3.3.4.1 IAR Memory Usage Details

The image below shows the IAR compiler optimization was used to generate the Memory Usage Details section.

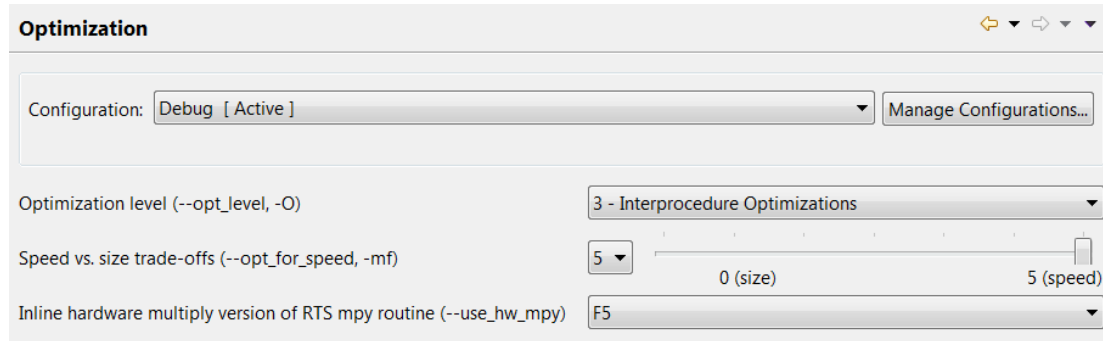


The IAR Example project requires a total of 31KB of FRAM and 4.3KB of SRAM, specifically the 4.3KB SRAM that is shared with LEA RAM. Table below has the breakup of memory usage by the msp-expfr5994\_gas\_afe\_bp example project.

MSP430FR5994 Example Project Memory Usage			
Memory Type	Code/Data	Size (Bytes)	Comments
FRAM	Code	15,920	
FRAM	Data	14,780	
SRAM	Code	0	
SRAM	Data	548	
SRAM	Data	300	Stack Usage
LEA-SRAM	Data	3,732	

### 1.3.3.4.2 CCS Memory Usage Details

The image below shows the CCS compiler optimization was used to generate the Memory Usage Details section.



The CCS Example project requires a total of 38.8KB of FRAM and 4.3KB of SRAM, specifically the 4.3KB SRAM that is shared with LEA RAM. Table below has the breakup of memory usage by the msp-expfr5994\_gas\_afe\_bp example project.

MSP430FR5994 Example Project Memory Usage			
Memory Type	Code/Data	Size (Bytes)	Comments
FRAM	Code	23,036	
FRAM	Data	15,748	
SRAM	Code	0	
SRAM	Data	562	
SRAM	Data	300	Stack Usage
LEA-SRAM	Data	3,732	

## 1.4 Supported IDEs

The following tool chains are supported:

- Texas Instruments Code Composer Studio v6.1.3 or later
- IAR Embedded Workbench for MSP430 v6.40.2 or later

## 2 Configuration API

[API Functions](#) .....17

### 2.0.1 Configuration\_api

#### Functions

[GASMETERING\\_return\\_code](#) [GASMETERING\\_configureUltrasonicMeasurement](#) ([GASMETERING\\_configuration](#) \*pLibraryConfig)

#### 2.0.1.1 Detailed Description

#### 2.0.1.2 Function Documentation

##### 2.0.1.2.1 **GASMETERING\_return\_code GASMETERING\_configureUltrasonicMeasurement (**

**GASMETERING\_configuration \* pLibraryConfig )** Configures measurement for first time.

This function is used to configure the ultrasonic measurement. This must be called before calling [GASMETERING\\_startUltrasonicMeasurement](#). This function will also initialize the GAS Metering handle.

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

Returns [GASMETERING\\_return\\_code](#)



## 3 Measurement API

API Functions ..... 19

### 3.0.2 Measurement\_api

#### Functions

[GASMETERING\\_return\\_code](#)      [GASMETERING\\_generateLPMDelay](#)      ([GASMETERING\\_configuration](#) \*pLibraryConfig, uint16\_t delayCount, [GASMETERING\\_low\\_power\\_mode](#) mode)

[GASMETERING\\_return\\_code](#)      [GASMETERING\\_startUltrasonicMeasurement](#)      ([GASMETERING\\_configuration](#) \*pLibraryConfig)

[GASMETERING\\_return\\_code](#)      [GASMETERING\\_updateUltrasonicMeasurement](#)      ([GASMETERING\\_configuration](#) \*pLibraryConfig)

#### 3.0.2.1 Detailed Description

#### 3.0.2.2 Function Documentation

##### 3.0.2.2.1 [GASMETERING\\_return\\_code](#) [GASMETERING\\_generateLPMDelay](#) (

[GASMETERING\\_configuration](#) \* pLibraryConfig,

uint16\_t delayCount,

[GASMETERING\\_low\\_power\\_mode](#) mode ) Generates a low power delay.

The following function allows user to generate a low power delay. The will remain in low power mode 3 until the timer expires. The clock source for this timer is ACLK/8. ACLK must the sources by a 32768 Hz crystal. Hence, the effective timer frequency for this timer is 4096 HZ. Minimum delay is  $1/4096\text{Hz} = 244 \text{ usec}$  and maximum delay is  $65536/4098\text{Hz} = 15.99 \text{ sec}$

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

*delayCount* Specifies count for low power delay. Valid values (0x0001 - 0xFFFF);

---

*mode* Specifies the low power mode to use for the delay

---

Returns [GASMETERING\\_return\\_code](#)

##### 3.0.2.2.2 [GASMETERING\\_return\\_code](#) [GASMETERING\\_startUltrasonicMeasurement](#) (

[GASMETERING\\_configuration](#) \* pLibraryConfig ) Triggers a firing and capture.

This function is used to trigger a firing/capture of ultrasonic signals. If an application interrupt occurs during the execution of this API, the capture of the signal might be corrupted. Therefore, user must ensure all application level interrupts are disabled before invoking this API.

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

Returns GASMETERING\_return\_code

### 3.0.2.2.3 GASMETERING\_return\_code GASMETERING\_updateUltrasonicMeasurement (

**GASMETERING\_configuration \* pLibraryConfig )** Updates measurement configuration during run time.

This function allows user to update library configuration during runtime.

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

Returns GASMETERING\_return\_code

## 4 Algorithms API

API Functions .....21

### 4.0.3 Algorithm\_api

#### Functions

`GASMETERING_return_code GASMETERING_initializeAlgorithm (GASMETERING_configuration *pLibraryConfig)`

`GASMETERING_return_code GASMETERING_runAlgorithm (GASMETERING_configuration *pLibraryConfig, GASMETERING_Algorithms_Results *pResults)`

#### 4.0.3.1 Detailed Description

#### 4.0.3.2 Function Documentation

##### 4.0.3.2.1 **GASMETERING\_return\_code GASMETERING\_initializeAlgorithm (**

**GASMETERING\_configuration \* pLibraryConfig )** Initializes algorithm.

This API must be called before calling GASMETERING\_runAlgorithm API. Failure to call this function might result in unknown behavior.

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

Returns GASMETERING\_return\_code

##### 4.0.3.2.2 **GASMETERING\_return\_code GASMETERING\_runAlgorithm (**

**GASMETERING\_configuration \* pLibraryConfig,**

**GASMETERING\_Algorithms\_Results \* pResults )** Runs ToF algorithm.

This API allows user to obtain Absolute Time of Flight, Delta ToF and volume flow rate. The have control on which calculation are run based on the selection of the following members of GASMETERING\_Algorithms\_Configuration structure:

isRunAbsToFUpEnabled

isRunAbsToFDownEnabled

isRunDeltaToFUpEnabled

isRunFlowEnabled

Parameters *pLibraryConfig* pointer to GAS Metering library configuration structure

---

*pResults* pointer to result structure.

---

Returns GASMETERING\_return\_code



---

# IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

## Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>

## Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © , Texas Instruments Incorporated