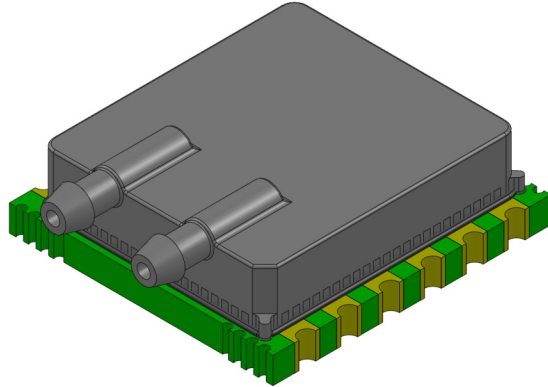




高精度複合型圧力センサ AUAVシリーズ



目次

特長およびアプリケーション	2
標準圧力範囲	2
圧力センサ最大定格	2
環境仕様	2
電気ブロック図	2
性能仕様	3
I2C / SPI 電氣的パラメーター	4
圧力出力伝達関数例	4
温度出力伝達関数	4
機能概要	5
I2C / SPI インターフェイスの概要	6-7
インターフェイス・タイミング図	8
デジタルインターフェイスコマンドフォーマット	9-11
I2C SPI コマンドシーケンス	12
初期化インストラクション	12
追加補償の説明	13-16
パッケージ図面	17
パッドレイアウト	17
製品識別	17
オーダーガイド	18

Introduction

オールセンサーズの複合機能センサファミリーのAUAVは、無人航空機（UAV）の厳しい要件に合わせて特別に設計されています。

当社の超高精度差圧センサダイと絶対圧センサダイを1つの強化されたプラットフォームへ収納することにより、優れた温度補正と直線性補正を実現しました。この画期的な技術によりAUAVは高精度の対気速度と高度測定が可能となりました。

自動化されたSMT PCBアセンブリプロセスと互換性のある非常にコンパクトなPCBモジュールの開発により小型化が実現されました。

飛行制御システムとのインターフェースは、I2CインターフェースとSPIインターフェースの両方が利用可能で、簡単に扱うことが可能です。

複数のチューブ・ポート・オプションがあり、2つのチューブ接続しか必要としない薄型サイドポート構成も含まれています。



For All Sensors Corporation's most recent quality certification documents, please visit www.allsensors.com



AUAV 高精度複合型圧力センサシリーズ

特長

- 表面実装
- I2CまたはSPIインターフェイス
- 広い温度範囲
- 高精度対気速度・高度測定

アプリケーション

- 無人航空機

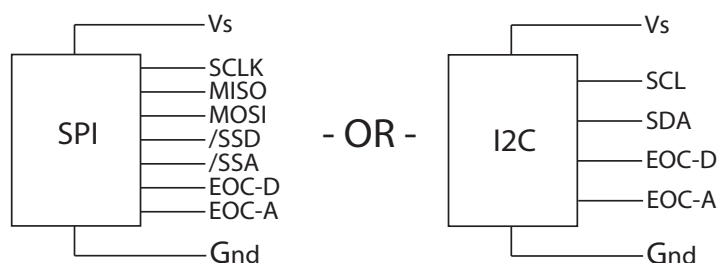
標準圧力範囲

デバイス	差圧	絶対圧	保証耐力		バースト圧力	
	動作範囲 ^A	動作範囲 ^A	inH2O	kPa	inH2O	kPa
L05D-M25125	± 5 inH2O	250mbarA - 1250 mbarA	270	67	415	103
L10D-M25125	± 10 inH2O	250mbarA - 1250 mbarA	300	75	415	103
L30D-M25125	± 30 inH2O	250mbarA - 1250 mbarA	350	87	415	103

Note A: その他の圧力レンジについては、お問い合わせ下さい。

圧力センサ最大定格		環境仕様	
供給電圧 (Vs)	3.63 Vdc	温度範囲	
コモンモード圧力	15 psig	補償:	-40°C to 85°C
SMTデバイス温度	245°C	動作:	-40°C to 85°C
		保管:	-40°C to 125°C
		湿度限界(結露なきこと)	0 to 95% RH

電気ブロック図



AUAVシリーズ高精度圧カセンサの性能特性

すべてのパラメータは、特に指定のない限り、 $\pm 3.3V \pm \%$ 励起、25°Cで測定。(注9: 差圧測定はポートHに正圧を印加した状態。絶対圧測定はポートLを参照。)

Parameter	Symbol	Min	Typ	Max	Units	Notes					
Output Span											
Differential Channel	Span _{dig}	-	$\pm 0.4 * 2^{24}$	-	Dec Count	1					
Absolute Channel	Span _{dig}	-	$0.8 * 2^{24}$	-	Dec Count	1					
Offset Output @ Zero Diff./ Abs. Pressure											
Differential Channel	OS _{dig}	-	$0.5 * 2^{24}$	-	Dec Count	-					
Absolute Channel	OS _{dig}	-	$0.1 * 2^{24}$	-	Dec Count	-					
Error Summary											
L05D - Differential Channel											
Total Error Band [85C to -20C]	TEB _P	-	TBD	± 0.25	%FSS	2, 6					
Total Error Band [-20C to -40C]	TEB _P	-	TBD	± 0.45	%FSS	2, 6					
Accuracy	Acc _P	-	TBD	± 0.10	%FSS	3, 6					
L10D - Differential Channel											
Total Error Band [85C to -20C]	TEB _P	-	TBD	± 0.20	%FSS	2, 6					
Total Error Band [-20C to -40C]	TEB _P	-	TBD	± 0.40	%FSS	2, 6					
Accuracy	Acc _P	-	TBD	± 0.10	%FSS	3, 6					
L30D - Differential Channel											
Total Error Band [85C to -20C]	TEB _P	-	TBD	± 0.15	%FSS	2, 6					
Total Error Band [-20C to -40C]	TEB _P	-	TBD	± 0.35	%FSS	2, 6					
Accuracy	Acc _P	-	TBD	± 0.10	%FSS	3, 6					
Absolute Channel - All variants											
Total Error Band [85C to -20C]	TEB _P	-	TBD	± 0.25	%FSS	2, 6					
Total Error Band [-20C to -40C]	TEB _P	-	TBD	± 0.45	%FSS	2, 6					
Accuracy	Acc _P	-	TBD	± 0.10	%FSS	3, 6					
Offset Position Sensitivity ($\pm 1g$)	Sen _{Pos}	-	± 0.10	-	%FSS	-					
Offset Long Term Drift (one year)	LTOS	-	± 0.25	-	%FSS	-					
Pressure Digital Resolution - No Missing Codes	Res _{Padc}	17.2	TBD	-	bit	-					
Temperature Output											
Resolution (internal)	Res _{Tadc}	-	13	-	bit	-					
Overall Accuracy	Acc _T	-	2	-	°C	-					
Electrical Specification:											
Supply Current Requirement						5, 7, 8					
During Active State	ICC _{Active}	-	5.3	7.3	mA	-					
During Idle State	ICC _{Idle}	-	1.7	7	μ A	-					
Power On Delay	t _{pwron}	-	-	2.5	ms	5					
Memory Read Access Time	t _{rd}	-	100	2000	μ s	4					
Data Update Time	t _{du}	(see table below)					5, 7				
Channel	Measurement Command										Units
	Single		Average2		Average4		Average8		Average16		
	Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max	
Absolute	4.9	5.4	9.6	10.6	19.0	21.1	37.8	41.9	75.2	83.3	ms
Differential	2.0	2.2	3.8	4.2	7.4	8.2	14.6	16.2	29.0	32.1	ms

仕様に関する注意事項

- 注1: スパンは、フル・スケールの小数点カウントとオフセットの小数点カウントの代数的な差です。フル・スケール圧力は校正された最大正圧です。
- 注2: トータル・エラー・バンドは、オフセットおよびスパンの温度誤差、校正誤差、直線性誤差、圧力ヒステリシス誤差、オフセット・ウォームアップ・シフト、オフセット位置感度、長期オフセット・ドリフト誤差から構成されます。
- 注3: 精度には、25°Cで評価された圧力ヒステリシス、再現性、ベストフィット直線性が含まれます。
- 注4: メモリ読み出し要求通信終了からメモリ・データ読み出し通信開始までの時間。
- 注5: パラメータは特性であり、100%テストされたものではありません。
- 注6: 追加補償セクションに記載されている以下の補正を評価。
- 注7: データ更新時間は、コマンド受信からビジー状態終了までの通信時間を除く。これは、アブソリュート・チャンネルではeocピンのLow状態時間として、差動チャンネルではeocパルスまでの遅延として観測される。
- 注8: 平均電流は、以下のように見積ることができる: ICC_{Idle} + (t_{DU} / READING INTERVAL) * ICC_{ACTIVE}。センサのアクティブおよびアイドル状態については、図2を参照してください。
- 注9: センサは3.3V電源で校正されていますが、内部レギュレーターにより、全体的な仕様に影響を与えることなく1.68V~3.6Vの電源電圧を使用することができます。これにより、バッテリー電源からの直接操作が可能になります。

info@all-device.com

https://all-device.com/

Fax : 047-489-5940

電話 : 047-489-5939

オールデバイス株式会社



I2C / SPI 電氣的パラメーター

Parameter	Symbol	Min	Typ	Max	Units	Notes
Input High Level	-	70	-	100	% of VS	5
Input Low Level	-	0	-	30	% of VS	5
Output Low Level (at 3mA sink)	-	-	-	10	% of VS	5
I2C Pull-Up Resistor	-	1000	-	-	Ω	5
I2C Load Capacitance on SDA, @ 400 kHz	C _{SDA}	-	-	200	pF	5
I2C Input Capacitance (each pin)	C _{I2C_IN}	-	-	10	pF	5
I2C Address:						
Differential Channel	Iadr _{Diff}	-	38	-	dec	-
Absolute Channel	Iadr _{Abs}	-	39	-	dec	-

圧力出力伝達関数 - 例

$$\text{Differential Pressure} = 1.25 \times \left(\frac{P_{dig} - 2^{23}}{2^{24}} \right) \times \text{Cal Range}$$

Where:

P_{dig} is the sensor 24-bit output, following corrections applied by extended compensation
 $CalRange$ is 2x the calibrated differential pressure range: eg. for L05D is 10 inH20

$$\text{Absolute Pressure} = 250 \text{ mbar} + 1.25 \times \left(\frac{P_{dig} - (0.1 \times 2^{24})}{2^{24}} \right) \times 1000 \text{ mbar}$$

Where:

P_{dig} is the sensor 24-bit output, following corrections applied by extended compensation

温度出力伝達関数

$$\text{Temperature } (^{\circ}\text{C}) = \left(\frac{Tout_{dig} * 155}{2^{24}} \right) - 45$$

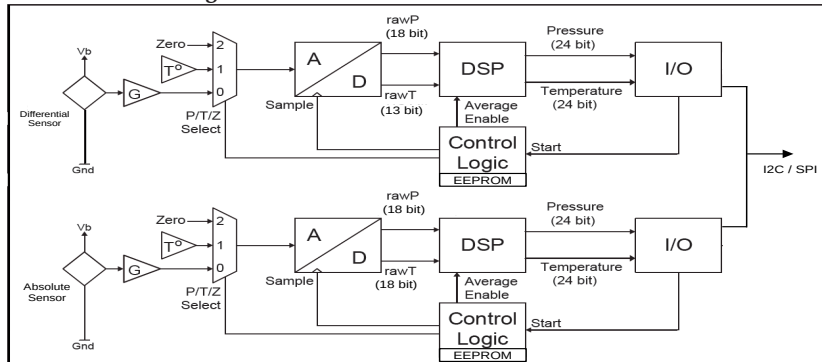
Where:

$Tout_{(dig)}$ The sensor 24-bit digital temperature output.

機能概要

AUAVは、2つの独立した信号経路を持つデジタルセンサで、それぞれにセンシングエレメント、差圧アンプ、アナログ・デジタルコンバーター、DSP、I2CまたはSPIインターフェースをサポートするIOブロックが含まれています（下図1参照）。センサの各チャンネルには、設定された動作モードをサポートするための内部温度基準と関連する制御ロジックも含まれています。ADC のフロント・エンドのマルチプレクサは、変換のための信号源を選択します。

Figure 1 - AUAV Block Diagram



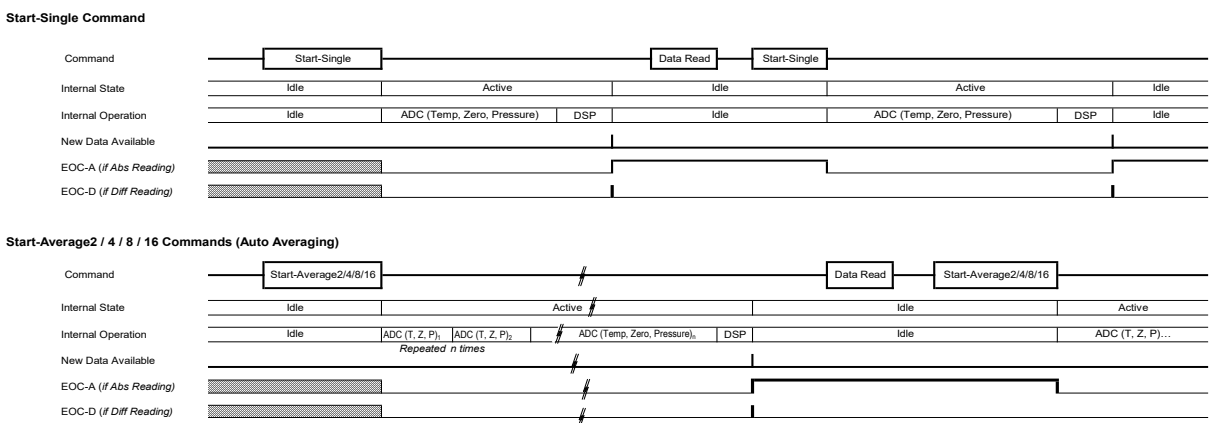
ADC は ADC 測定サイクルの間、Raw センサ信号 (P)、温度リファレンス (T)、ゼロ・リファレンス (Z) に対して変換を実行します。

DSPは変換された圧力と温度の情報を受け取り、圧力出力を補正するために複数の伝達関数を適用します。この伝達関数には、スパン、オフセット、スパンに対する温度効果、オフセットに対する温度効果、オフセットに対する2次温度効果の補正が含まれます。この補正された出力は、「追加補償命令」のセクションで後述するように、追加の外部補正を適用することでさらに改善されます。

センサ・コマンド： 5つの測定コマンドがサポートされており、1つの圧力/温度の読み取り値、または2、4、8、16の読み取り値の平均値を返します。これらの各コマンドは、センサをアイドル状態からアクティブ状態にし、測定サイクルを開始します。Start-Averageコマンドの場合、このサイクルは適切な回数繰り返され、Start-Singleコマンドは1回の反復を実行します。DSPが計算を完了し、新しい値がI/Oブロックに提供されると、センサはアイドル状態に戻る。センサは次の測定コマンドを受信するまで、この低電力状態を維持します。AUAVの各チャンネルはこのように独立して動作するため、システムのデータ収集設計に柔軟性があります。

測定完了後、Data Read コマンドを使用して結果を読み取ることができます。ADC と DSP はアイドル状態のまま、I/O ブロックは 7 バイトのステータスと測定データを返します（図 2 を参照）。いつでも、ホストは Status Read コマンドで現在のデバイスのステータスを要求することができます。全てのコマンドの概要については表 1 を参照してください。

Figure 2 - AUAV Communication Model



I2C / SPI インターフェ이스の概要

I/Oインターフェース設定： センサは、以下のプロトコルに基づき、SPI または I2C シリアルインターフェースを自動的に選択します： SS入力がホストによってLowに設定されると（SPIコマンド・トランザクション中に発生）、電源が切れるまでI/OインターフェースはSPI通信用に設定されたままになります。それ以外の場合、有効なデバイス・アドレスとコマンドがI2Cインターフェースで受信されると、電源が切れるまでI/OインターフェースはI2C用に設定されたままになります。

詳細については、初期化手順（12ページ）を参照してください。

I2Cバス通信の概要

2線式I2Cインターフェースは、通信に標準的な信号シーケンスを使用する。タイミング要件は図 5 に示す。

バス・ノット・ビジー (I) : アイドル期間中、データ線 (SDA) とクロック線 (SCL) は共にHIGHのままである。

START 条件 (ST) : クロック(SCL)がHIGHの間にSDA線がHIGHからLOWに遷移することをSTART条件と解釈する。START条件は常にマスターによって設定される。各コマンド・リクエストはスタート・コンディションで始まらなければならない。

スレーブ・アドレス (An) : I2C バスでは各デバイスに固有のアドレスが必要です。AUAV センサには、各チャンネルのスレーブアドレスがあらかじめ設定されています（「仕様」セクションで定義済み）。START 条件を設定した後、マスターは 7 ビットのセンサアドレスを含むアドレスバイトとデータ方向ビット (R/W) を送信します。0 "はマスタからスレーブへの送信 (WRITE) を示し、"1 "はデバイスからマスタへの要求 (READ) を示します。

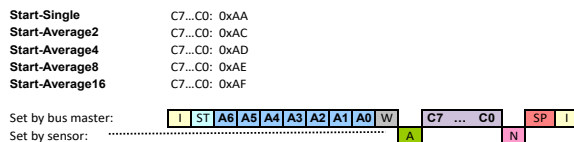
アックレッジ (AまたはN) : データは一度に8ビット (1バイト) 単位でMSBファーストで転送される。データを受信する各デバイスは、マスタであれスレーブであれ、データを受信を確認するためにデータ・ラインをLOWにプルする必要があります。マスターは、この目的のために余分なクロックパルスを生成しなければならない。レシーバーがデータ・ラインをプルダウンしない場合、NACK状態が存在し、スレーブ・トランスミッターは非アクティブになる。マスターは マスターは、最後のコマンドを再度送信するか、STOP 条件を設定して転送を終了するかを決定する。

DATA valid (Dn) : START 条件の後、クロック信号(SCL)のHIGH 期間データ線が安定しているとき、データ線の状態は有効なデータを表す。データ線SDAはSCLのLOW期間中に変更されなければならない。SCLの立ち上がりエッジでSDAのデータ値が取り込まれ、1データビットあたり1クロックパルスとなる。

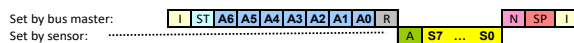
STOP 条件 (P) : クロック(SCL)がHIGHの間にSDA線がLOWからHIGHに遷移すると停止状態を示す。STOP条件は常にマスターによって生成される。

Figure 3 - I2C Communication Diagram

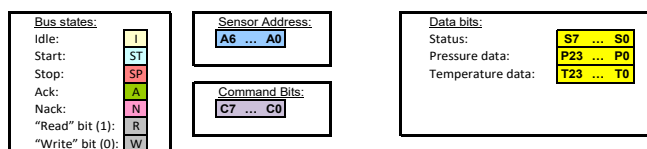
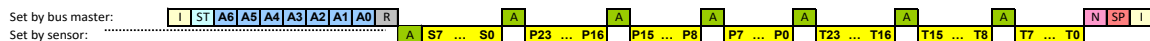
1. Measurement Commands: Start-Single (to start reading of single sample):



2. Status Read:



3. Data Read:

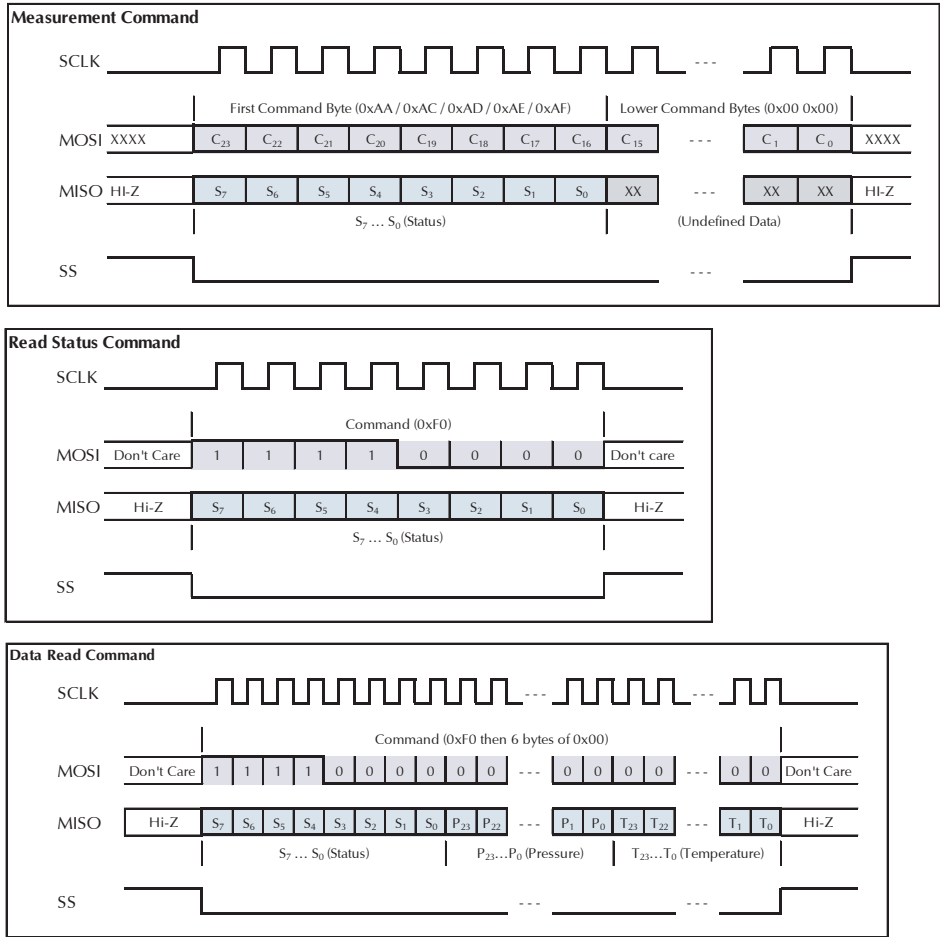


I2C / SPI インターフェイスの概要 (続き)

SPI バス通信の概要

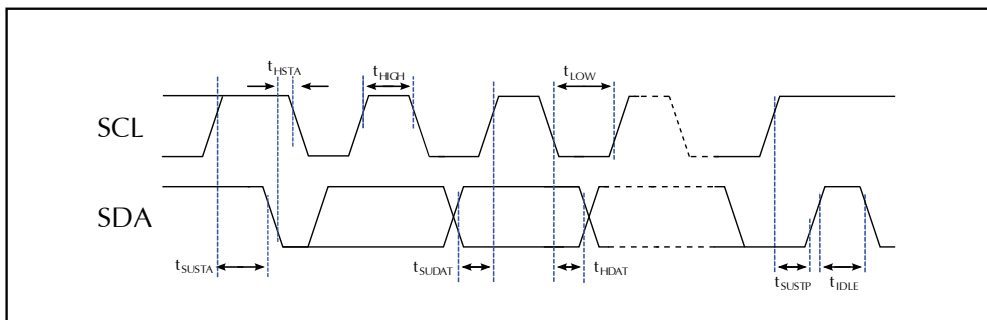
ビットとバス信号のシーケンスを下図 (図4) に示す。詳細なタイミング・データについては、インターフェース・タイミング図のセクションの図5を参照してください。

Figure 4 - SPI Communications Diagram



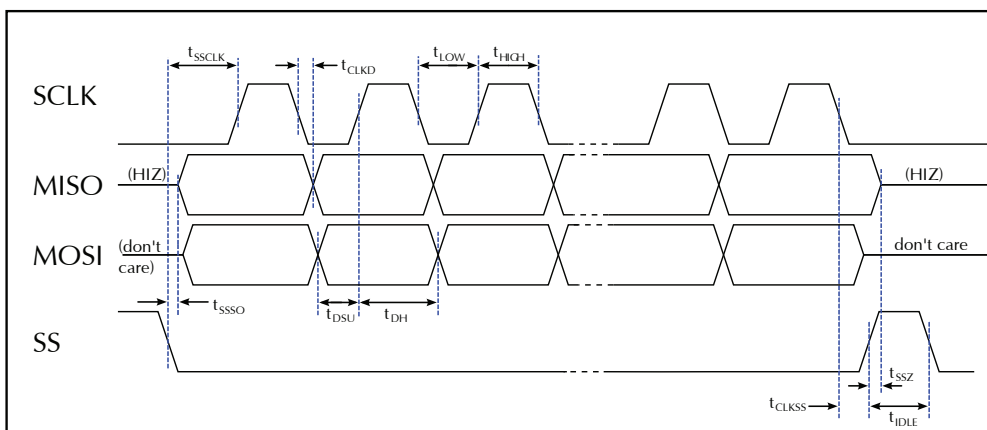
インターフェイスタイミング図

Figure 5 - I2C Timing Diagram



PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
SCL frequency	f_{SCL}	100	-	400	KHz
SCL low width	t_{LOW}	1.3	-	-	us
SCL high width	t_{HIGH}	0.6	-	-	us
Start condition setup	t_{SUSTA}	0.6	-	-	us
Start condition hold	t_{HSTA}	0.6	-	-	us
Data setup to clock	t_{SUDAT}	0.1	-	-	us
Data hold to clock	t_{HDAT}	0	-	-	us
Stop condition setup	t_{SUSTP}	0.6	-	-	us
Bus idle time	t_{IDLE}	2.0	-	-	us

Figure 6 - SPI Timing Diagram



PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
SCLK frequency	f_{SCLK}	0.05	1	5	MHz
SCLK duty cycle [$t_{HIGH}/(t_{LOW}+t_{HIGH})$]	D_{SCLK}	40	50	60	%
/SS low to first rising clock edge	t_{SSCLK}	5	50	--	ns
/SS low to MISO enabled	t_{SSSO}	--	--	20	ns
/SS high to MISO disabled	t_{SSZ}	--	--	20	ns
Clock to MISO data out	t_{CLKD}	8	--	20	ns
MOSI setup to rising clock	t_{DSU}	20	--	--	ns
MOSI hold after rising clock	t_{DH}	20	--	--	ns
Last falling clock to rising /SS	t_{CLKSS}	0	--	--	ns
Bus idle time	t_{IDLE}	10	--	--	μ s

機能概要

デジタル・インターフェイス・コマンド・フォーマット

測定の開始または EEPROM データの読み取りを要求する場合、I2C のコマンド長は 1 バイト、SPI のコマンド長は 3 バイトです。

I2C でセンサの状態を要求する場合、ホストは単純に 1 バイトの読み取り転送を実行します。

SPI でセンサのステータスを要求する場合、ホストは 1 バイトを読み出しながら Status Read コマンドバイトを送信する必要があります。I2C 経由でセンサ・データを読み出す場合、ホストは単純に 7 バイト・リード転送を実行します。

SPI 経由でセンサ・データを読み出す場合、ホストはデータを読み出しながら 7 バイトの Data Read コマンドを送信する必要があります。

文書化されていないコマンドをセンサに送信すると、キャリブレーションが破損します。保証の対象外となります。

測定コマンド、センサデータ読み取り、センサステータス読み取りの詳細については、以下の表 1 を参照してください。AUAV センサの各チャンネルは、I2C アドレスまたは SPI/SS ラインで選択されたコマンドを独立して処理することに注意してください。

Table 1 - AUAV Sensor Reading Command Set

Measurement Commands				
Description	SPI (3 bytes)			I2C (1 byte)
Start-Single	0xAA	0x00	0x00	0xAA
Start-Average2	0xAC	0x00	0x00	0xAC
Start-Average4	0xAD	0x00	0x00	0xAD
Start-Average8	0xAE	0x00	0x00	0xAE
Start-Average16	0xAF	0x00	0x00	0xAF

Read Sensor Data	
I2C	Read of 7 bytes from device
SPI	Read of 7 bytes from device Host must send [0xF0], then 6 bytes of [0x00] on MOSI Sensor Returns 7 bytes on MISO

Read Sensor Status	
I2C	Read of 1 byte from device.
SPI	Read of 1 byte from device Host must send [0xF0] on MOSI Sensor Returns 1 byte on MISO

デジタル・インターフェイス・コマンド・フォーマット(続き)

メモリ読み出しコマンドは、センサの内部メモリから拡張補償係数を読み出すために使用されます。拡張補償のセクションで詳しく説明するように、これらは A、B、C、D、Es、TC50H、TC50L と呼ばれます。値 (A、B、C、D) は 32 ビット符号付き整数で、各チャンネルの EEPROM 内の 8 つの 16 ビットレジスタに格納されています。値 Es、TC50H、TC50L は 8 ビット符号付き整数で、2 つの 16 ビット EEPROM 位置に格納されています。各チャンネルのメモリ・マップは表2に示されています：

Table 2 - Coefficient Memory Map

Absolute Channel:

Address	47 (0x2F)	48 (0x30)	49 (0x31)	50 (0x32)	51 (0x33)	52 (0x34)	53 (0x35)	54 (0x36)	55 (0x37)	56 (0x38)
Coeff. Word	[AHW] A (High Word)	[ALW] A (Low Word)	[BHW] B (High Word)	[BLW] B (Low Word)	[CHW] C (High Word)	[CLW] C (Low Word)	[DHW] D (High Word)	[DLW] D (Low Word)	[TC50H] [TC50L]	[0] [Es]

Differential Channel:

Address	43 (0x2B)	44 (0x2C)	45 (0x2D)	46 (0x2E)	47 (0x2F)	48 (0x30)	49 (0x31)	50 (0x32)	51 (0x33)	52 (0x34)
Coeff. Word	[AHW] A (High Word)	[ALW] A (Low Word)	[BHW] B (High Word)	[BLW] B (Low Word)	[CHW] C (High Word)	[CLW] C (Low Word)	[DHW] D (High Word)	[DLW] D (Low Word)	[TC50H] [TC50L]	[0] [Es]

これらの値を読み出すための一連のコマンドは、メモリ・リード・リクエスト (表3参照) の後に、メモリ・データ・リード (表4参照) が続く形となる。読み出し要求と読み出し動作の間には、メモリ読み出しアクセス時間遅延を守らなければならないことに注意すること。

Table 3 - Memory Read Request Command

Memory Commands: I2C or SPI:			
Description	Command (3 bytes)		
Read Request <i>Absolute Channel</i>	<EEPROM Address> (Values 47 -56 only)	0x00	0x00
<i>Differential Channel</i>	(Values 43 -52 only)		

これらのコマンドを正確かつ慎重に使用することを強調しなければならない。これらのコマンドの形成や送信を誤ると、較正データが上書きされ、センサの動作が低下する可能性があります。

Table 4 - Memory Data Read Operation

Read Memory Data	
I2C	Read of 3 bytes from device.
SPI	Read of 3 bytes from device. Host must send [0xF0], then 2 bytes of [0x00] on MOSI. Sensor returns 3 bytes on MISO.

Example : I2C Read of Coefficient **B** from Absolute channel EEPROM :

Write <0x31> , and read back: <Status> <BHW>. (high 16 bits of B, as 2 bytes)

Write <0x32> , and read back: <Status> <BLW>. (low 16 bits of B, as 2 bytes)

B = [BHW:BLW], assembling BHW and BLW into a signed 32-bit integer.

Example : SPI Read of Coefficient **D** from Differential channel EEPROM :

Write <0x31><0x00><0x00>, over SPI MOSI output.

Set output buffer to <0xF0><0x00><0x00>, then perform 3-byte transfer.

Input buffer will then contain: <Status> <DHW_(high byte)> <DHW_(low byte)>.

Write <0x32><0x00><0x00>, over SPI MOSI output.

Set output buffer to <0xF0><0x00><0x00>, then perform 3-byte transfer.

Input buffer will then contain: <Status> <DLW_(high byte)> <DLW_(low byte)>.

D = [DHW:DLW], assembling DHW and DLW into a signed 32-bit integer.

デジタル・インターフェイス・データ・フォーマット

いずれのタイプのデジタル・インターフェイスでも、センサから返されるデータの形式は同じです。測定データの場合、最初のバイトはステータス・バイトで構成され、その後24ビットの符号なし圧力値と24ビットの符号なし温度値が続きます。圧力と温度への変換については、圧力出力伝達関数と温度出力伝達関数の定義を参照してください。センサ出力には、拡張補正調整を使用して除去しなければならないエラー項が含まれていることに注意してください。これらの計算については、「拡張補正命令セクション」を参照してください。

メモリ・データの読み取りでは、ステータス・バイトに続いて、メモリ・ワードの上位バイト、下位バイトが読み取られます。

センサの測定データ形式については表 5 を参照。表 6 に EEPROM リード・データの形式を、表 7 にステータス・バイトの定義を示します。

エラーなしで読み取りが完了すると、絶対値チャンネルの場合は 0x40、差動チャンネルの場合は 0x50 のステータスが返されることに注意してください。

Table 5 - Measurement Output Data Format

S[7:0]	P[23:16]	P[15:8]	P[7:0]	T[23:16]	T[15:8]	T[7:0]
Status Byte	Pressure Byte 2	Pressure Byte 1	Pressure Byte 0	Temperature Byte 2	Temperature Byte 1	Temperature Byte 0

Table 6 - Memory Data Output Format

S[7:0]	MEM[15:8]	MEM[7:0]
Status Byte	MEM High Byte	MEM Low Byte

Table 7- Status Byte Definition

Bit	Description
Bit 7 [MSB]	[Always = 0]
6	Power: [1 = Power On]
5	Busy: [1 = Processing Command, 0 = Ready]
4:3	Mode - Absolute Channel: [00 = Normal Operation, others = Command Fault] Mode - Differential Channel: [10 = Normal Operation, others = Command Fault]
2	Memory Error: [1 = EEPROM Checksum Fail]
1	Sensor Configuration: [always = 0]
Bit 0 [LSB]	ALU Error: [1 = Error: Pressure Overrange or Underrange]

I2C SPI コマンドシーケンス

I2Cコマンドシーケンス

電源投入後、製品はアイドル状態になり、バス・マスターからのコマンドを待つ。表 1 に示すように、5 つの測定コマンドのいずれかを送信できます。これらのコマンド・バイトの 1 つを受信すると、EOC-A ピンがロー・レベルに設定され（アブソリュート・チャネルのみ）、ステータス・バイトにセンサ・ビジー・ビットが設定されます。アクティブ状態で測定と計算が完了すると、補正されたデータが出力レジスタに書き込まれ、デジタル・インターフェイスからアクセスできるようになります。アブソリュート・チャネルの場合、EOC-A ピンはハイ・レベルに設定され、次のコマンドを受信するまでハイ・レベルのままです。差動チャネルの場合、EOC-Dに5マイクロ秒のハイレベル・パルスが設定されます。どちらのチャネルで、BUSYビットがクリアされ、処理コアはアイドル状態に戻る。その後、ホスト・プロセッサは、I2Cの場合は単純に7バイトのデバイス・リードであるデータ・リード動作を実行できる。EOC ピンがモニターされていない場合、ホストは Status Read コマンドを繰り返すことでステータス・バイトをポーリングすることができる。ステータス・バイトのビジー・ビットがゼロの場合、有効なデータが準備できていることを示し、7バイト全てのデータ・リードを実行することができる。

表 1、3、4 で定義されている以外のコマンドをセンサーに送信しないでください。

SPI コマンドシーケンス

I2C インターフェイス・コンフィギュレーションと同様に、製品は電源投入後にアイドル状態になり、SPI マスターからのコマンドを待ちます。測定サイクルを開始するには、3 バイトの測定コマンド（表 1 を参照）のいずれかをマスタから発行する必要があります。メモリ読み出し動作を開始するには、メモリ読み出し要求（表3を参照）を送信する必要があります。このコマンド要求中にセンサから返されるデータは、ステータス・バイトの後に続く 2 つの未定義データ・バイトで構成されます。

測定コマンドのデコードに成功すると、アブソリュート・チャネルのみ、EOC-A ピンが Low に設定され、コアは測定と計算のためにアクティブ状態になります。完了すると、更新されたセンサ・データが出力レジスタに書き込まれ、コアはアイドル状態に戻ります。絶対値チャネルの場合、EOC-Aピンはこの時点でHighレベルに設定され、ビジー・ステータス・ビットは0に設定されます。差動チャネルの場合、EOC-Dピンはこの時点で5マイクロ秒のHighパルスを供給し、ビジー・ステータス・ビットは0に設定されます。アクティブ期間中またはアイドル期間中の任意の時点で、SPIマスタはステータス・リード・コマンド（値 0xF0の1バイト）を送信してステータス・バイトを要求できます。

I2Cコンフィギュレーションと同様に、ステータス・バイトのビジー・ビット（値 0）（またはアブソリュート・チャネルのEOC-A ピンのハイレベル）は、有効なデータ・セットが センサから読み出される可能性があることを示します。データ読み出しコマンドは、SPI マスターから送信する必要があります（最初のバイトは値 0xF0 で、その後 6 バイトの 0x00 が続きます）。メモリ読み出し操作については、結果の読み出しについて表 4 を参照してください。

初期化インストラクション

あらゆる運転条件下で信頼できる運転を保証するために、以下のシステム設計項目を推奨します：

1. 高 EMI 環境で動作させる場合、電気設計には Vs と Test ピンの間に 10 K Ω ~47 K Ω のプルアップ抵抗を含める必要がある。
2. 電源投入後、SPI通信を行う場合、以下の信号シーケンスによりSPIへのインターフェイス・ロックが保証される：
 - a. 差動チャネルのEEPROMリードを実行する。
 - b. アブソリュート・チャネルで EEPROM リードを実行します。
3. EEPROMの読み取りを進め、拡張補償係数を取得する。
4. 高温では、EEPROMの最大読み出しアクセス時間を観察してください。
 - a. EEPROMの結果と共にBUSYステータスが返された場合は、読み取り要求から結果のリードバックまでの時間を長くして再試行してください。

追加補償の説明

AUAVシリーズセンサは、追加補償係数を含む内部メモリロケーションを持っています。圧力測定値の精度を最適化するために、システム設計者はこれらの値を使用して、センサから送信されるデータに追加の3次エラー補正調整を適用し、さらに温度補正を行うことができます。

追加補償の理論:

4つの直線性係数は、工場でセンサの各チャンネルに対して3次最小化解としてプログラムされる。

$$(1) \quad \text{Error} = \text{Pref} - (\text{POut} + f(\text{POut})),$$

where
Pref is the true pressure applied;
POut is the sensor output;
f(POut) is a cubic correction function, Ax^3+Bx^2+Cx+D .

Then

$$(2) \quad \text{Pcorr} = \text{Pout} + f(\text{Pout}) \text{ as the linearity-corrected pressure value.}$$

For improved accuracy over temperature, residual temperature dependent errors are minimized by the term:

$$(3) \quad \text{TCadj} = (1 - (\text{Es} * 2.5 * |0.5 - \text{Pcorr}|)) * (\text{T} - \text{Tref}) * \text{TC50}$$

where:

$$\text{TC50} = \text{TC50H}/\text{TC50Scale} \text{ for } \text{T} > \text{Tref}$$
$$\text{TC50} = \text{TC50L}/\text{TC50Scale} \text{ for } \text{T} < \text{Tref}$$

and

$$\text{TC50Scale} = 100 * 100 * 167772.$$

This represents the possible range of temperature-dependent error, scaled to temperature counts.

Then

$$(4) \quad \text{Pcomp} = \text{Pcorr} - \text{TCadj}$$

for the final optimized pressure value. This is used in the Pressure Output Transfer Function on Page 4 to obtain pressure in appropriate units.

実装：使用方法:

システム起動時:

データシート「Digital Interface Command Formats」セクションに記載されているコマンド・シーケンスを使用して、各チャンネルのセンサEEPROMから7つの係数（A、B、C、D、Es、TC50H、TC50L）を読み込む。A、B、C、Dは32ビット符号付き整数で、-1.0～+1.0のスケールされた大きさを表します。Es、TC50H、TC50Lは8ビット符号付き整数で、-1.0から+1.0までのスケールされたマグニチュードを表します。

コード例:

Declarations:

```
extern unsigned char inbuf[], outbuf[]; // sensor
float ALIN_A , ALIN_B, ALIN_C, ALIN_D, A_Es, A_TC50H, A_TC50L; // Abs coeffs
float DLIN_A , DLIN_B, DLIN_C, DLIN_D, D_Es, D_TC50H, D_TC50L; // Diff coeffs

// Platform-dependent SPI fns, which perform EEPROM read of specified
// address and following address. These 16b values are merged as [HW:LW]
// for an int32 return.
extern int32_t Read_ACfg_SPI(uint8_t regadr);
extern int32_t Read_DCfg_SPI(uint8_t regadr);

void Init_Aconfig(void);
void Init_Dconfig(void);
```

追加補償の説明（続き）

センサ電源投入後、EEPROM 係数値を読み出し、符号付き 32 ビット整数を Float に変換する：

絶対チャンネルの場合：

```
void Init_Aconfig(void)
{
    int32_t i32A = 0, i32B = 0, i32C = 0, i32D=0, i32TC50HLE=0;
    int8_t i8TC50H = 0, i8TC50L = 0, i8Es = 0;

    // These i32 Reads return 2 register values merged as int32
    // i32 then normalized to +/- 1.0

    i32A = Read_ACfg_SPI(47); // returns 47 | 48 as int32
    ALIN_A = ((float) (i32A)) / ((float) (0x7FFFFFFF));

    i32B = Read_ACfg_SPI(49); // returns 49 | 50 as int32
    ALIN_B = (float) (i32B) / (float) (0x7FFFFFFF);

    i32C = Read_ACfg_SPI(51); // returns 51 | 52 as int32
    ALIN_C = (float) (i32C) / (float) (0x7FFFFFFF);

    i32D = Read_ACfg_SPI(53); // returns 53 | 54 as int32
    ALIN_D = (float) (i32D) / (float) (0x7FFFFFFF);

    i32TC50HLE = Read_ACfg_SPI(55); // 55: TC50H | TC50L
    i8TC50H = (i32TC50HLE >> 24) & 0xFF; // 55 H
    i8TC50L = (i32TC50HLE >> 16) & 0xFF; // 55 L
    i8Es = (i32TC50HLE) & 0xFF; // 56 L

    A_Es = (float) (i8Es) / (float) (0x7F);
    A_TC50H = (float) (i8TC50H) / (float) (0x7F);
    A_TC50L = (float) (i8TC50L) / (float) (0x7F);
}
```

差動チャンネル関数 Init_Dconfig() も同様ですが、下位レベルの関数 Read_DCfg_SPI() で EEPROM アドレス 43 ~ 52 を使用し、DLIN_*、D_Es、D_TC50H、D_TC50L を初期化します。

これらのA、B、C、D、Es、TC50H、TC50Lの浮動小数点定数値は、次のように各読み取りに適用される調整で使用されます：

各読み取りに適用される補正：

センサから読み取った各圧力値 (POut) について、以下を計算します。

$$PComp = POut + A*POut^3 + B*POut^2 + C*POut + D - TCadj.$$

アプリケーションが常に25℃付近の環境で動作する場合、TCadjはオプションの用語である。

例：I2Cを使用した差動チャンネルの読み取り：

差動センサの読み取りを開始する：

```
outbuf[0] = 0xAD; ui8Address = 38; // Request 4x oversampled reading
// Platform-dependent I2C write function, for example
I2C_Write(ui8Address, outbuf, 1); // send 1-byte request
```

変換遅延後（またはEOC-Dパルス受信時）、結果を読み出す：

```
I2C_Read(ui8Address, inbuf, 7); // read 7 bytes: Status, P, T
```

追加補償の説明 (続き)

センサの 24 ビット圧力値と温度値が、ステータス・バイトに先行して I2C 入力バッファに格納される。つまり、inbuf[0]=ステータス・バイト、inbuf[1]、[2]、[3]には圧力値、inbuf[4]、[5]、[6]には温度が格納されます。

次に、差動チャンネル EEPROM から読み出した係数を使用して、補正調整を適用します：

Declarations & definitions:

```
// constants:
const int32_t Tref_Counts = 7576807; // temperature counts at 25C
const float TC50Scale = 100.0 * 100.0 * 167772.2 // scale TC50 to 1.0% FS0

// local variables:
float AP3, BP2, CP, Corr, Pcorr, Pdiff, TCadj, TC50, Pnfso, Tcorr, Pcorrt;
int32_t iPraw, Tdiff, iTemp, iPCorrected;
uint32_t PComp;
```

First, correct for linearity, as in Equation (2):

```
// Convert unsigned 24-bit pressure value to signed +/- 23-bit:
iPraw = (inbuf[1]<<16) + (inbuf[2]<<8) + inbuf[3] - 0x800000;
// Convert signed 23-bit value to float, normalized to +/- 1.0:
Pnorm = (float)iPraw; // cast to float
Pnorm /= (float) 0x7FFFFFFF;

AP3 = DLIN_A * Pnorm * Pnorm * Pnorm; // A*Pout^3
BP2 = DLIN_B * Pnorm * Pnorm; // B*Pout^2
CP = DLIN_C * Pnorm; // C*Pout
Corr = AP3 + BP2 + CP + DLIN_D; // Linearity correction term
Pcorr = Pnorm + Corr; // Corrected P, range +/-1.0.
```

この時点で、Pcorrは25°Cの温度で最適化された直線性補正圧力を表します。室温アプリケーションでは、これで十分な補正が可能です。

センサのネイティブ・フォーマットである符号なし24ビット整数に変換して戻します：

```
iPcorr = (int32_t)(Pcorr * (float)0x7FFFFFFF); // Convert to signed 23-bit
iPcorr += 0x800000; // Back to unsigned 24-bit
```

その後、Pdigの値としてiPcorrを使用して、4ページの圧力出力伝達関数を適用することができます。

25°Cを大幅に上回ったり下回ったりする温度にさらされるシステムでは、以下に示すように追加の温度補正を適用することができます：

```
// Compute difference from reference temperature, in sensor counts:
iTemp = (inbuf[4]<<16) + (inbuf[5]<<8) + inbuf[6]; // 24-bit temperature
Tdiff = iTemp - Tref_Counts; // see constant defined above.
```

Re-normalize the linearity-corrected pressure from +/- 1.0 to [0 - 1.0):

```
Pnfso = (Pcorr + 1.0)/2.0;

//TC50: Select High/Low, based on current temp above/below 25C:
if (Tdiff > 0)
    TC50 = D_TC50H;
else
    TC50 = D_TC50L;
// Find absolute difference between midrange and reading (abs(Pnfso-0.5)):
if (Pnfso > 0.5)
    Pdiff = Pnfso - 0.5;
else
    Pdiff = 0.5 - Pnfso;
```

追加補償の説明（続き）

ここで、圧力と基準値からの温度差の関数として、式（3）のような温度依存性の調整を行う：

$$T_{corr} = (1.0 - (D_{Es} * 2.5 * P_{diff})) * T_{diff} * TC50 / TC50Scale;$$

this adjustment is applied to the linearity-corrected value, as in Equation(4):

$$P_{Corrt} = P_{nfso} - T_{corr}; // \text{corrected } P: \text{float, [0 to +1.0]}$$

Re-normalize back to unsigned 24-bit value, finishing Equation (4):

$$P_{comp} = (\text{uint32_t}) (P_{Corrt} * (\text{float})0\text{xFFFFFF});$$

Start next sensor reading:

```
outbuf[0] = 0xAD; // Request 4x oversampled reading
success = I2C_Write(ui8Address, outbuf, 1) // send 1-byte request
```

圧力単位に変換：

P_{comp}の結果はセンサの補正された出力を表し、校正された測定単位で圧力を計算するときにP_{dig}として圧力出力伝達関数で使用されます。

例として、差動チャンネルの例を続けます：
差動校正レンジが +/- 10 inH₂Oのセンサーの場合、

$$P_{inH2O} = 1.25 * ((P_{dig} - 2^{23}) / 2^{24}) * (2 * 10) \text{ inH}_2\text{O}$$

If a reading from this sensor results in P_{comp} = 9145890 counts,

$$P_{inH2O} = 1.25 * ((9145890 - 8388608) / 16777216) * 20 \text{ inH}_2\text{O} \\ = 1.1284 \text{ inH}_2\text{O}$$

For a compensated reading of P_{comp} = 3000000 counts,

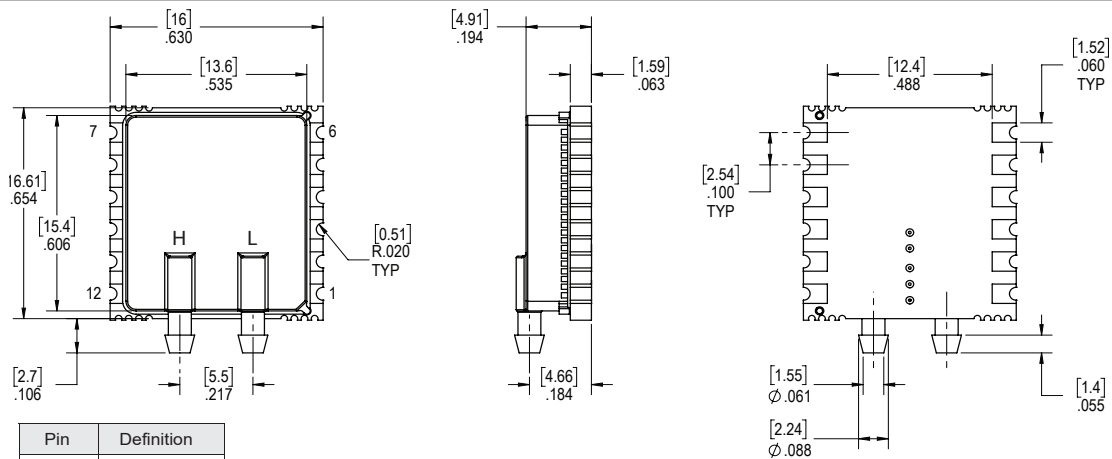
$$P_{inH2O} = 1.25 * ((3000000 - 8388608) / 16777216) * 20 \text{ inH}_2\text{O} \\ = -8.0297 \text{ inH}_2\text{O}$$

アブソリュート・チャンネルでは、P_{comp}を得るための計算は同じであるが、係数 ALIN_A、ALIN_B、ALIN_C、ALIN_D、A_Es、A_TC50H、およびA_TC50Lのセットを使用する。

この伝達関数は、差動チャンネルの伝達関数とはオフセット値が異なる、
例えば) 9145890カウントのP_{comp}値は絶対圧を表します：

$$P_{mbarA} = 250 \text{ mbarA} + 1.25 * (P_{dig} - (0.1 * 2^{24}) / 2^{24}) * 1000 \text{ mbarA} \\ P_{mbarA} = 250 \text{ mbarA} + 1.25 * ((9145890 - 1677722) / 16777216) * 1000 \text{ mbarA} \\ = 806.42 \text{ mbarA}$$

パッケージ図



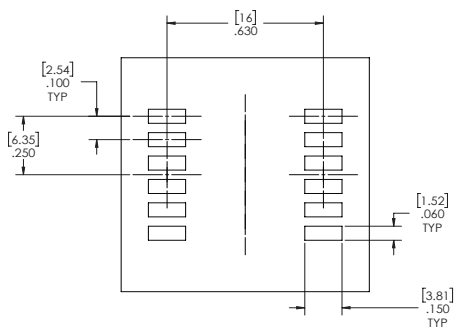
Pin	Definition
1	SCL/SCLK
2	SDA/MOSI
3	EOC-A
4	/SSA
5	GND
6	Vs
7	N/C
8	GND
9	TEST
10	EOC-D
11	/SSD
12	MISO

備考 (特に指定のない限り)

- 1) 寸法はインチ[mm]単位です。
- 2) TEST端子：EMIイミュニティのためVsにプルアップ。

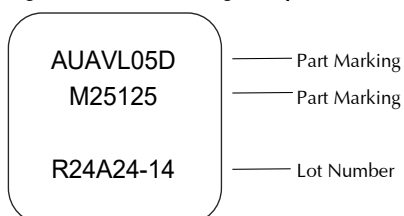
標準ポートオプション
追加ポート・オプションについては、ご相談下さい。

パッドレイアウト



製品識別

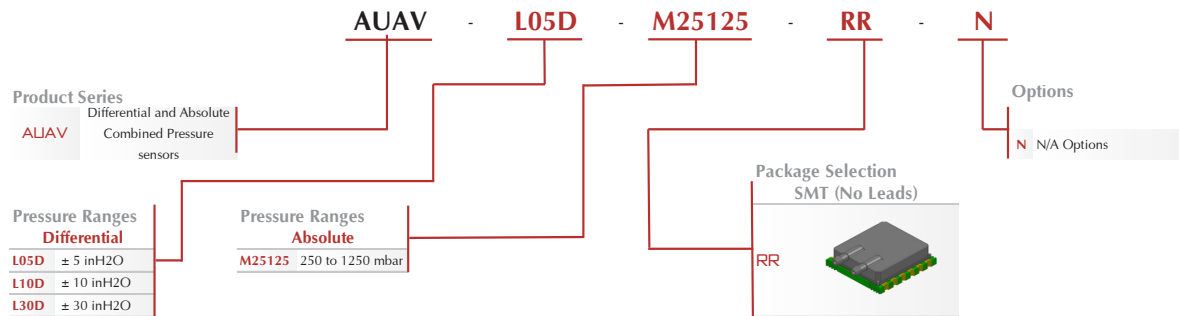
Figure 7: Product Marking Example



All Sensors	
TITLE:	AUAV Package
SIZE	FILE NAME
A	AUAV Package

オーダーガイド AUAVシリーズ

たとえば、**AUAV-L05D-M25125-RR-N** は、差圧5 inH₂O、絶対圧範囲250~1250 mbar、RRパッケージ（SMT、2つのバンプ付きサイドポート、コーティングオプションなし）のAUAVシリーズ圧力センサを定義しています。



All Sensors reserves the right to make changes to any products herein. All Sensors does not assume any liability arising out of the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.