ALL SENSORS®
an Amphenol company

DLLR - High Accuracy Pressure Sensors Series

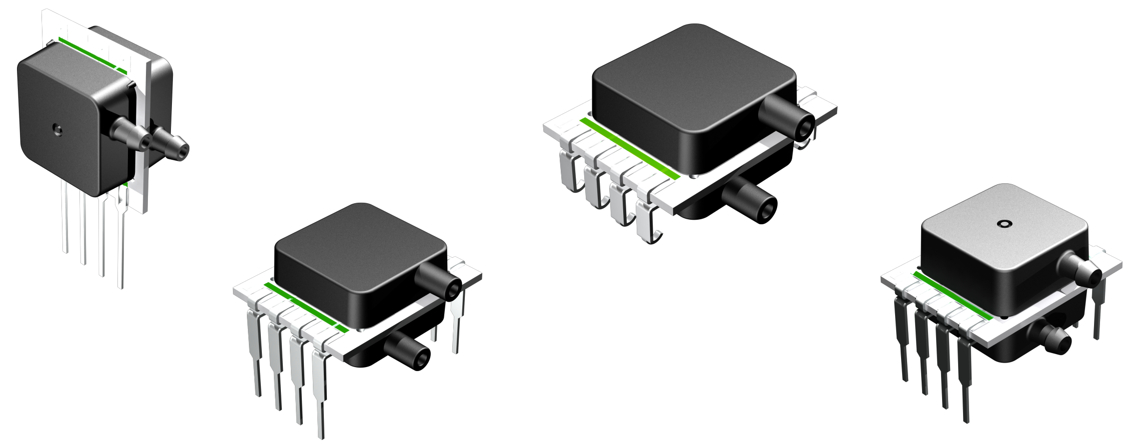


Table of Contents

Introduction

The DLLR Series Mini Digital Output Sensor is based on All Sensors' CoBeam² ™ Technology. This reduces package stress susceptibility, resulting in improved overall long term stability and vastly improves the position sensitivity.

Extra compensation terms programmed into each sensor allow system firmware to remove high-order linearity errors, providing extremely accurate pressure linearity. For applications exposed to changing temperatures, additional terms can be applied for very low-temperature related errors.

The digital interface eases integration of the sensors into a wide range of process control and measurement systems, allowing direct connection to serial communications channels. For battery-powered systems, the sensors can enter very low-power mode between readings to minimize load on the power supply.

These calibrated and compensated sensors provide accurate, stable output over a wide temperature range. This series is intended for use with non-corrosive, non-ionic working fluids such as air, dry gases.

https://www.allsensors.com/products/dllr-series

All Sensors Corporation's Quality Management System has been certified by TUV SUD in accordance with the ISO 9001:2015 Standard.

A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com ALL SENSORS

Features

- Pressure Ranges of 10 & 30 inH2O
- 1.68V to 3.6V Supply Voltage Range
- I2C or SPI Interface (Automatically Selected)
- Better Than 0.10% Accuracy
- High Resolution 16/17/18 Bit Output
- Oversampling Options for Noise Reduction

Applications

- Medical Breathing
- Environmental Controls
- HVAC
- Industrial Controls
- Portable / Hand-Held Equipment

Standard Pressure Ranges

Device	Operating Range [A]		Proof Pressure		Burst Pressure		Nominal Span
	inH2O	Pa	inH2O	kPa	inH2O	kPa	Counts
DLLR-L10D	±10	±2488.4	100	25	300	75	$\pm 0.4 * 2^{24}$
DLLR-L10G	0 to 10	2488.4	100	25	300	75	$0.8 * 2^{24}$
DLLR-L30D	±30	±7465.2	200	50	300	75	$\pm 0.4 * 2^{24}$
DLLR-L30G	0 to 30	7465.2	200	50	300	75	$0.8 * 2^{24}$

Note A: Operating range in Pa is expressed as an approximate value.

Pressure Sensor Maximum Ratings

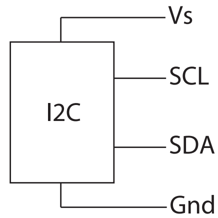
Supply Voltage (Vs)	3.63 Vdc
Common Mode Pressure	10 psig
Lead Temperature (soldering 2-4 sec.)	270°C

Environmental Specifications

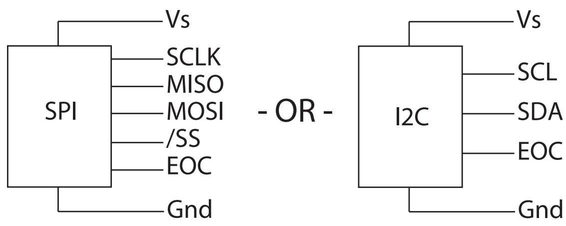
Temperature Ranges	
Compensated: Commercial	0°C to 70°C
Operating	-25°C to 85°C
Storage	-40°C to 125°C
Humidity Limits (non condensing)	0 to 95% RH

Electrical Block Diagram

For SIP Packages



For DIP and J-Lead Packages



Performance Characteristics for DLLR Series High Accuracy Low Pressure Sensors

All parameters are measured at ±3.3V ±5% excitation and 25C unless otherwise specified (Note 9). Pressure measurements are with positive pressure applied to PORT B.

Parameter	Min	Typ	Max	Units	Notes
Output Span					
LxxD	-	$\pm 0.4 * 2^{24}$	-	Dec Count	1
LxxG	-	$0.8 * 2^{24}$	-	Dec Count	1
Offset Output @ Zero Diff. Pressure (OS_{dig})					
LxxD	-	$0.5 * 2^{24}$	-	Dec Count	-
LxxG	-	$0.1 * 2^{24}$	-	Dec Count	-
Error Summary					
L10D					
Total Error Band	-	±0.10	±0.25	%FSS	2, 6
Span Temperature Shift	-	±6	-	ppmFSS/C	4, 6
Offset Temperature Shift	-	±9	-	ppmFSS/C	4, 6
Accuracy	-	±0.03	±0.10	%FSS	3, 6
L10G					
Total Error Band	-	±0.06	±0.20	%FSS	2, 6
Span Temperature Shift	-	±7	-	ppmFSS/C	4, 6
Offset Temperature Shift	-	±3	-	ppmFSS/C	4, 6
Accuracy	-	±0.03	±0.10	%FSS	3, 6
L30D					
Total Error Band	-	±0.10	±0.35	%FSS	2, 6
Span Temperature Shift	-	±10	-	ppmFSS/C	4, 6
Offset Temperature Shift	-	±4	-	ppmFSS/C	4, 6
Accuracy	-	±0.03	±0.10	%FSS	3, 6
L30G					
Total Error Band	-	±0.05	±0.15	%FSS	2, 6
Span Temperature Shift	-	±6	-	ppmFSS/C	4, 6
Offset Temperature Shift	-	±3	-	ppmFSS/C	4, 6
Accuracy	-	±0.03	±0.10	%FSS	3, 6
Offset Position Sensitivity (±1g)	-	±0.10	-	%FSS	-
Offset Long Term Drift (one year)	-	±0.25	-	%FSS	-
Pressure Digital Resolution - No Missing Codes					
16-bit Option	15.7	-	-	bit	-
17-bit Option	16.7	-	-	bit	-
18-bit Option	17.7	-	-	bit	-
Temperature Output					
Resolution	-	16	-	bit	-
Overall Accuracy	-	2	-	°C	-
Supply Current Requirement					5, 7, 8
During Active State (ICC_{Active})	-	2	2.6	mA	-
During Idle State (ICC_{Idle})	-	100	250	nA	-
Power On Delay	-	-	2.5	ms	5
Memory Read Access Time	30	-	-	ms	10
Data Update Time (t_{DU})		(see table below)			5, 7

Calibrated Resolution	Measurement Command										Units
	Single		Average2		Average4		Average8		Average16		
	Typ	Max	Typ	Max	Typ	Max	Typ	Max	Typ	Max	
16 bit option	2.80	3.1	5.40	6.0	10.60	11.7	21.00	23.2	41.80	46.0	ms
17 bit option	3.20	3.6	6.20	6.9	12.20	13.5	24.20	26.7	48.20	53.1	ms
18 bit option	3.70	4.1	7.20	8.0	14.20	15.7	28.20	31.1	56.20	61.9	ms

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

I2C / SPI Electrical Parameters for DLLR Series

Parameter	Symbol	Min	Typ	Max	Units	Notes
Input High Level	-	80	-	100	% of Vs	5
Input Low Level	-	0	-	20	% of Vs	5
Output Low Level	-	-	-	10	% of Vs	5
I2C Pull-up Resistor	-	1000	-	-	Ω	5
I2C Load Capacitance on SDA, @ 400 kHz	CSDA	-	-	200	pF	5
I2C Input Capacitance (each pin)	CI2C_IN	-	-	10	pF	5

Pressure Output Transfer Function

$$Pressure(inH_2O) = 1.25 \times \left(\frac{dig - OS_{dig}}{2^{24}}\right) \times FSS(inH_2O)$$

Where:

dig Is the sensor 24-bit digital output, following corrections applied by extended compensation.

OS_{dig} Is the specified digital offset

For Gage Operating Range sensors: $0.1 * 2^{24}$

For Differential Operating Range sensors: $0.5 * 2^{24}$

$FSS(inH_2O)$ The sensor Full Scale Span in inches H_2O

For Gage Operating Range sensors: Full Scale Pressure

For Differential Operating Range sensors: 2 x Full Scale Pressure

Temperature Output Transfer Function

$$emperature\ (°C) = \left(\frac{out_{dig} * 125}{2^{24}}\right) - 40$$

Where:

out_{dig} The sensor 24-bit digital temperature output.

Specification Notes

NOTE 1: THE SPAN IS THE ALGEBRAIC DIFFERENCE BETWEEN FULL SCALE DECIMAL COUNTS AND THE OFFSET DECIMAL COUNTS. THE FULL SCALE PRESSURE IS THE MAXIMUM POSITIVE CALIBRATED PRESSURE.

NOTE 2: TOTAL ERROR BAND CONSISTS OF OFFSET AND SPAN TEMPERATURE AND CALIBRATION ERRORS, LINEARITY AND PRESSURE HYSTERESIS ERRORS, OFFSET WARM-UP SHIFT, OFFSET POSITION SENSITIVITY AND LONG TERM OFFSET DRIFT ERRORS.

NOTE 3: ACCURACY INCLUDES PRESSURE HYSTERESIS, REPEATABILITY AND BEST-FIT STRAIGHT LINE LINEARITY, EVALUATED AT 25C.

NOTE 4: PARTS PER MILLION OF FULL-SCALE SPAN PER DEGREE C.

NOTE 5: PARAMETER IS CHARACTERIZED AND NOT 100% TESTED.

NOTE 6: EVALUATED FOLLOWING CORRECTIONS DESCRIBED IN EXTENDED COMPENSATION SECTION.

NOTE 7: DATA UPDATE TIME IS EXCLUSIVE OF COMMUNICATIONS, FROM COMMAND RECEIVED TO END OF BUSY STATUS. THIS CAN BE OBSERVED AS EOC PIN LOW- STATE DURATION.

NOTE 8: AVERAGE CURRENT CAN BE ESTIMATED AS : ICC_{Idle} + (t_{DU} / READING INTERVAL) * ICCACTIVE). REFER TO FIGURE 2 FOR ACTIVE AND IDLE CONDITIONS OF THE SENSOR (THE ACTIVE STATE IS WHILE EOC PIN IS LOW).

NOTE 9: THE SENSOR IS CALIBRATED WITH A 3.3V SUPPLY HOWEVER, AN INTERNAL REGULATOR ALLOWS A SUPPLY VOLTAGE OF 1.68V TO 3.6V TO BE USED WITHOUT AFFECTING THE OVERALL SPECIFICATIONS. THIS ALLOWS DIRECT OPERATION FROM A BATTERY SUPPLY.

NOTE 10: DELAY BETWEEN END OF MEMORY READ REQUEST COMMUNICATION AND START OF MEMORY DATA READ COMMUNICATION.

Device Ordering Options

Output Resolution

Calibrated output resolution can be ordered to be 16, 17, or 18 bits.

Higher resolution results in slower update times; see the Data Update Time in the Performance Characteristics table.

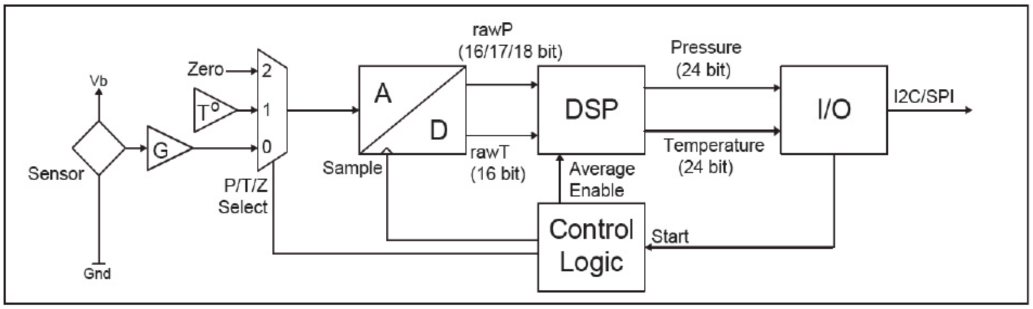
ALL SENSORS

A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

Operation Overview

The DLLR is a digital sensor with a signal path that includes a sensing element, a variable- bit analog to digital converter, a DSP and an IO block that supports either an I2C or SPI interface (see Figure 1 below). The sensor also includes an internal temperature reference and associated control logic to support the configured operating mode. Since there is a single ADC, there is also a multiplexer at the front end of the ADC that selects the signal source for the ADC.

Figure 1 - DLLR Block Diagram



The ADC performs conversions on the raw sensor signal (P), the temperature reference (T) and a zero reference (Z) during the ADC measurement cycle.

The DSP receives the converted pressure and temperature information and applies a multi-order transfer function to compensate the pressure output. This transfer function includes compensation for span, offset, temperature effects on span, temperature effects on offset and second order temperature effects on offset. There is also linearity compensation for gage devices and front to back linearity compensation for differential devices. This compensated output is further improved by applying additional external correction, as described later in the Extended Compensation instructions section.

Sensor Commands: Five Measurement commands are supported, returning values of either a single pressure / temperature reading or an average of 2, 4, 8, or 16 readings. Each of these commands wakes the sensor from Idle state into Active state, and starts a measurement cycle. For the Start-Average commands, this cycle is repeated the appropriate numper of times, while the Start-Single command performs a single iteration. When the DSP has completed calculations and the new values have been made available to the I/O block, the sensor returns to Idle state. The sensor remains in this low-power state until another Measurement command is received.

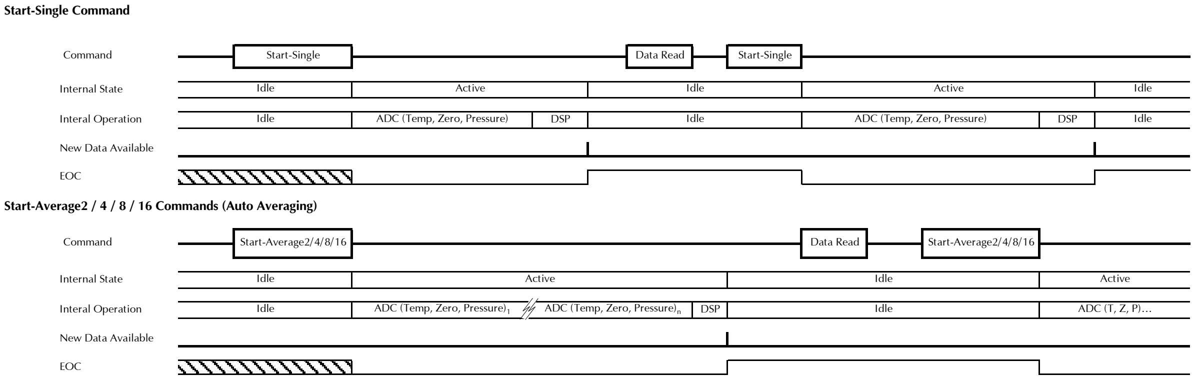
After completion of the measurement, the result may then be read using the Data Read command. The ADC and DSP remain in Idle state, and the I/O block returns the 7 bytes of status and measurement data. See Figure 2, following. At any time, the host may request current device status with the Status Read command.
See Table 1 for a summary of all commands.

For optimum sensor performance, All Sensors recommends that Measurement commands be issued at a fixed interval by the host system. Irregular request intervals may increase overall noise on the output.
Furthermore, if reading intervals are much slower than the Device Update Time, using the Averaging commands is suggested to reduce offset shift. This shift is constant with respect to time interval, and may be removed by the application. For longer fixed reading intervals, this shift may be removed by the factory on special request.

I/O Interface Configuration: The sensor automatically selects SPI or I2C serial interface, based on the following protocol: If the /SS input is set low by the host (as occurs during a SPI command transaction), the I/O interface will remain configured for SPI communications until power is removed. Otherwise, once a valid device address and command have been received over the I2C interface, the I/O interface will remain configured for I2C until power is removed.
NOTE: The four-pin (SIP) packages only support the I2C interface.

Operation Overview

Figure 2 - DLLR Communication Model



Digital Interface Command Formats

When requesting the start of a measurement, the command length for I2C is 1 byte, for SPI it is 3 bytes.
When requesting sensor status over I2C, the host simply performs a 1-byte read transfer.
When requesting sensor status over SPI, the host *MUST* send the Status Read command byte while reading 1 byte.
When reading sensor data over I2C, the host simply performs a 7-byte read transfer.
When reading sensor data over SPI, the host *MUST* send the 7-byte Data Read command while reading the data.

SENDING UNDOCUMENTED COMMANDS TO SENSOR WILL CORRUPT CALIBRATION AND IS NOT COVERED BY WARRANTY.

See Table 1 below for Measurement Commands, Sensor Data read and Sensor Status read details.

Table 1 - DLLR Sensor 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*

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

Digital Interface Command Formats (Cont'd)

The Memory Read Command is used to retrieve the extended Compensation Coefficients from internal memory of the sensor. Values (A, B, C, and D) are 32-bit signed integers, stored in eight 16-bit registers at addresses 47 through 54. Values TC50H and TC50L are stored in high byte and low byte, respectively, of address 55, as signed 8-bit integers. Value E is an 8-bit signed integer, stored at Low Byte of address 56.

Table 2 - Coefficient Memory Map

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

Each Word is stored in form ([High Byte]:[Low Byte]).
To form the complete integers A, B, C, and D, assemble the words in order ([xHW] : [xLW]). For E, the 8-bit low byte represents the complete integer. For TC50H and TC50L, the high byte and low byte, respectively, represent the complete integers.

The sequence of commands to retrieve these values is in the form of a Memory Read Request (See Table 3) followed by a Memory Data Read (See Table 4). Note that the Memory Read Access Time delay must be observed between the request and the read operations.

Table 3 - Memory Read Request Command

Memory Commands: I2C Write or SPI MOSI:				
Description	**SPI (3 bytes)**			**I2C (1 byte)**
Read Request	<EEPROM Address> (*Values 47 -56 only*)	0x00	0x00	<EEPROM Address> (*Values 47 -56 only*)

It must be emphasized that these commands be used accurately and carefully. Errors in forming or transmitting these commands can result in degraded sensor operation.

Table 4 - Memory Data Read Operation

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

*Example : I2C Read of Coefficient **B** :*

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** :*

Write <**0x35**><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**$_{\text{(high byte)}}$> < **DHW**$_{\text{(low byte)}}$>.
Write <**0x36**><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**$_{\text{(high byte)}}$> < **DLW**$_{\text{(low byte)}}$>.
D = [DHW:DLW], assembling DHW and DLW into a signed 32-bit integer.

Digital Interface Data Format

For either type of digital interface, the format of data returned from the sensor is the same. For measurement data, the first byte consists of the Status Byte followed by a 24-bit unsigned pressure value and a 24-bit unsigned temperature value. See the Pressure Output Transfer Function and Temperature Output Transfer Function definitions on page 3 for converting to pressure and temperature. Note that the sensor output includes error terms that must be removed in system software. Refer to *'Extended Compensation Instructions Section'* for these computations.

For memory data output, the status byte is followed by the high byte, then low byte of the memory word.

Refer to Table 5 for the measurement data format of the sensor. Table 6 shows the formation of EEPROM-read data, and Table 7 shows the Status Byte definition.

Note that a completed reading without error will return status 0x40.

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 3	Pressure Byte 1	Pressure Byte 0	Temperature Byte 3	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	4Mode: [00 = Normal Operation, others = Command Fault]
2	Memory Error [1 = EEPROM Checksum Fail] Sensor
1	Configuration [always = 0]
Bit 0 [LSB]	ALU Error [1 = Error: Pressure Overrange or Underrange]

I2C Interface

<u>I2C Command Sequence</u>

The part enters Idle state after power-up, and waits for a command from the bus master. Any of the five Measurement commands may be sent, as shown in Table 1. Following receipt of one of these command bytes, the EOC pin is set to Low level, and the sensor Busy bit is set in the Status Byte. After completion of measurement and calculation in the Active state, compensated data is written to the output registers, the EOC pin is set high, the BUSY bit is cleared and the processing core goes back to Idle state. The host processor can then perform the Data Read operation, which for I2C is simply a 7-byte Device Read.

If the EOC pin is not monitored, the host can poll the Status Byte by repeating the Status Read command, which for I2C is a one-byte Device Read. When the Busy bit in the Status byte is zero, this indicate that valid data is ready, and a full Data Read of all 7 bytes may be performed.

DO NOT SEND COMMANDS TO SENSOR OTHER THAN THOSE DEFINED IN TABLES 1, 3 & 4.

ALL SENSORS

A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

I2C Interface (Cont'd)

I2C Bus Communications Overview

The I2C interface uses a set of signal sequences for communication. The following is a description of the supported sequences and their associated mnemonics. Refer to Figure 3 for the associated usage of the following signal sequences.

Bus not Busy (I): During idle periods both data line (SDA) and clock line (SCL) remain HIGH.

START condition (ST): A HIGH to LOW transition of SDA line while the clock (SCL) is HIGH is interpreted as START condition. START conditions are always set by the master. Each initial request for a pressure value has to begin with a START condition.

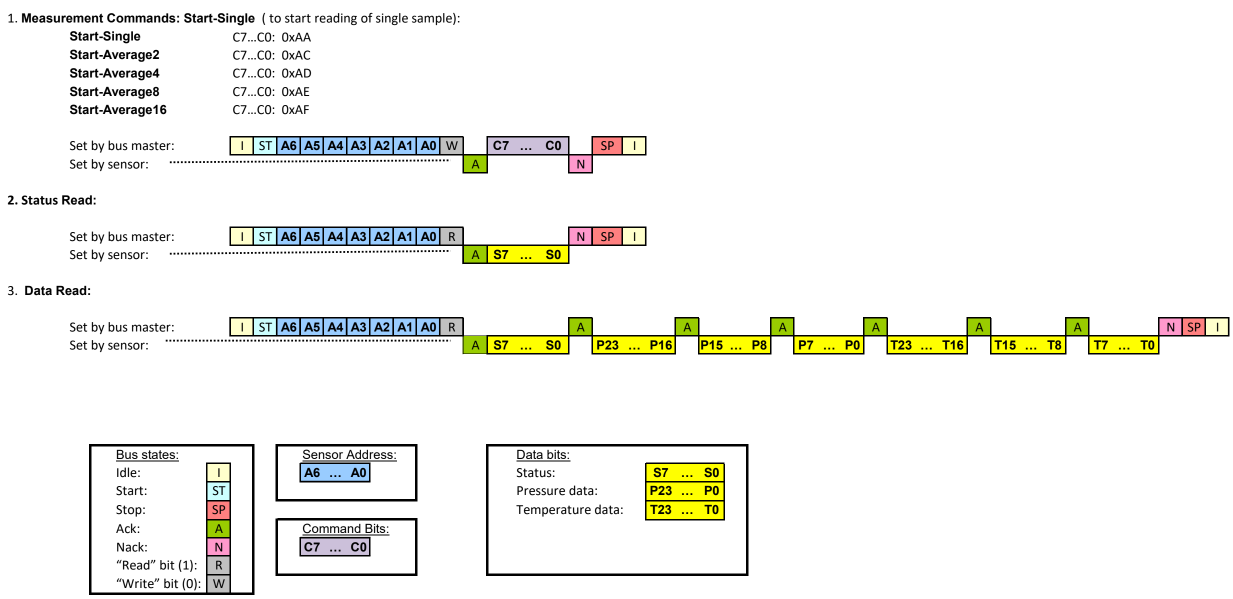
Slave address (An): The I2C-bus requires a unique address for each device. The DLLR sensor has a preconfigured slave address (defined by device option, see Table 9). After setting a START condition the master sends the address byte containing the 7 bit sensor address followed by a data direction bit (R/W). A "0" indicates a transmission from master to slave (WRITE), a "1" indicates a device-to master request (READ).

Acknowledge (A or N): Data is transferred in units of 8 bits (1 byte) at a time, MSB first. Each data-receiving device, whether master or slave, is required to pull the data line LOW to acknowledge receipt of the data. The Master must generate an extra clock pulse for this purpose. If the receiver does not pull the data line down, a NACK condition exists, and the slave transmitter becomes inactive. The master determines whether to send the last command again or to set the STOP condition, ending the transfer.

DATA valid (Dn): State of data line represents valid data when, after a START condition, data line is stable for duration of HIGH period of clock signal. Data on line must be changed during LOW period of clock signal. There is one clock pulse per data bit.

STOP condition (P): LOW to HIGH transition of the SDA line while clock (SCL) is HIGH indicates a STOP condition. STOP conditions are always generated by the master.

Figure 3 - I2C Communication Diagram



SPI Interface

SPI Command Sequence

As with the I2C interface configuration, the part enters Idle state after power-up, and waits for a command from the SPI master. To start a measurement cycle, one of the 3- byte Measurement Commands (see Table 1) must be issued by the master. To start a memory read operation, the memory read request (see Table 3) must be sent. The data returned by the sensor during this command request consists of the Status Byte followed by two undefined data bytes.

On successful decode of a measurement command, the EOC pin is set Low as the core goes into Active state for measurement and calculation. When complete, updated sensor data is written to the output registers, and the core goes back to the Idle state. The EOC pin is set to a High level at this point, and the Busy status bit is set to 0. At any point during the Active or Idle periods, the SPI master can request the Status Byte by sending a Status Read command (a single byte with value 0xF0).

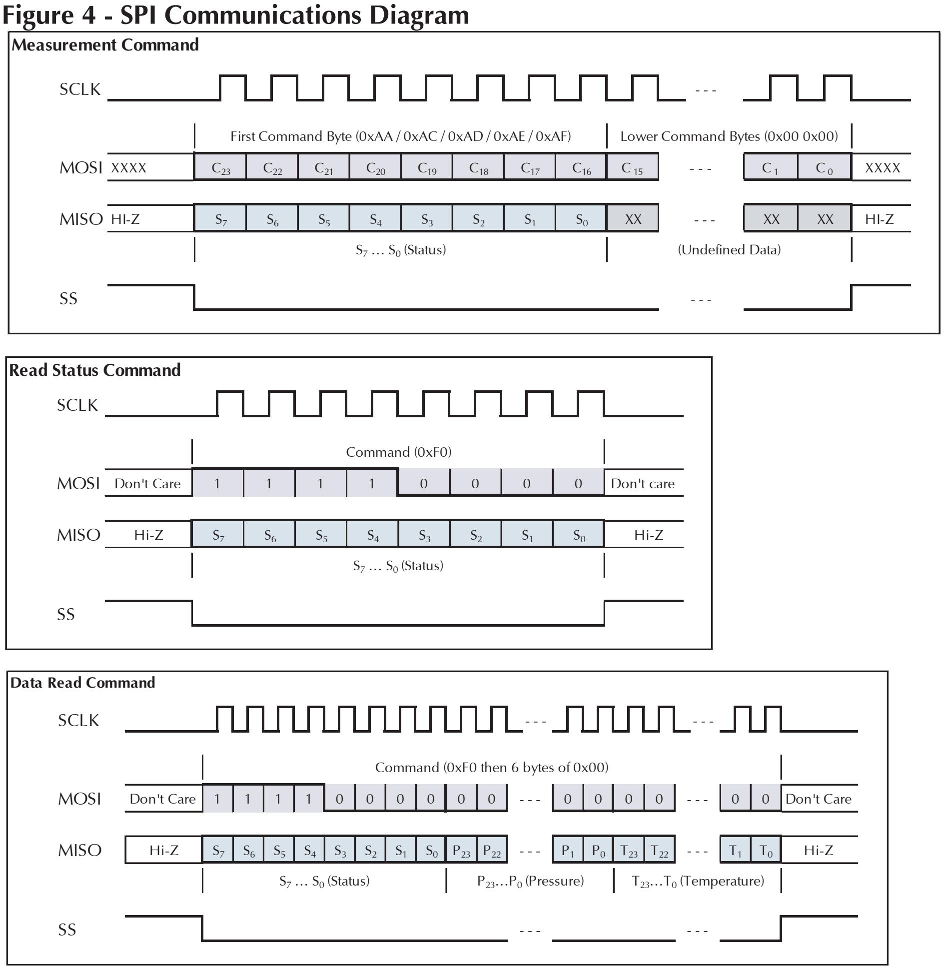
As with the I2C configuration, a Busy bit of value 0 in the Status Byte or a high level on the EOC pin indicates that a valid data set may be read from the sensor. The Data Read command must be sent from the SPI master (The first byte of value 0xF0 followed by 6 bytes of 0x00). For memory read operations, see Table 4 for reading back the result.

NOTE: Sending commands that are not defined in Tables 1, 3, or 4 will corrupt sensor operation.

SPI Bus Communications Overview

The sequence of bits and bus signals are shown in the following illustration (Figure 4). Refer to Figure 5 in the Interface Timing Diagram section for detailed timing data.

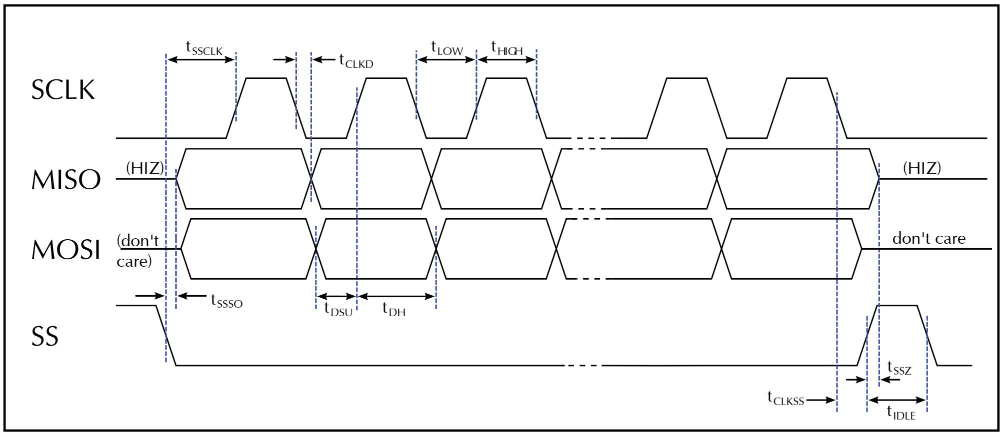
Figure 4 - SPI Communications Diagram



ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

Interface Timing Diagrams

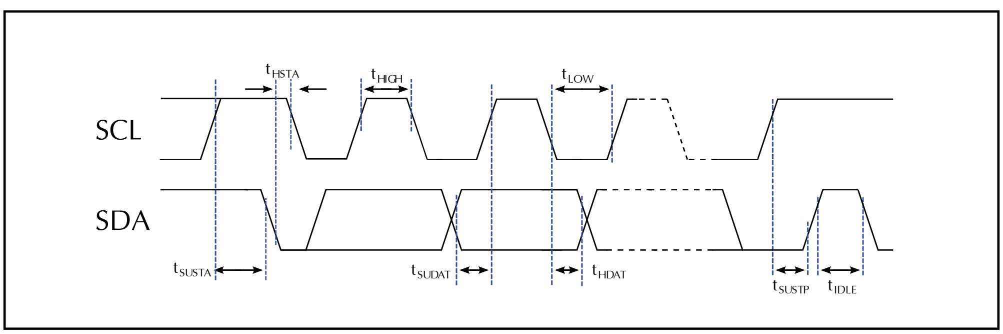
Figure 5 - SPI Timing Diagram



PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
SCLK frequency [(1)]	f_{SCLK}	0.05	-	5	MHz
SS low to first clock edge	t_{SSCLK}	120	-	-	ns
SS low to serial out	t_{SSSO}	--	-	20	ns
Clock to data out	t_{CLKD}	8	-	32	ns
SCLK low width	t_{LOW}	100	-	-	ns
SCLK high width	t_{HIGH}	100	-	-	ns
Data setup to clock	t_{DSU}	50	-	-	ns
Data hold after clock	t_{DH}	50	-	-	ns
Last clock to rising SS	t_{CLKSS}	0	-	-	ns
SS high to output hi-Z	t_{SSZ}	--	-	20	ns
Bus idle time	t_{IDLE}	250	-	-	ns

(1) Maximum by design, tested to 1.0 MHz.

Figure 6 - 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

Extended Compensation Instructions

DLLR Series sensors have internal memory locations containing extended compensation coefficients. For optimal accuracy of pressure readings, system designers can use these values to apply an additional 3rd-order error-correction adjustment to data delivered from the sensor, as well as additional temperature compensation.

Theory of Extended Compensation:
The four linearity coefficients are obtained for each sensor at the factory by a 3rd order minimization solution to

(1) **Error = Pref - (POut + f(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) **Pcorr = POut + f(POut)** as the linearity-corrected pressure value.

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

(3) **TCadj = (1 - (E * 2.5 *| 0.5 - Pcorr |)) * (T - Tref) * TC50**

where:

TC50 = TC50H/TCKscale for T > Tref
TC50 = TC50L/TCKscale for T < Tref

and:

TCKscale = 100 * 100 *134218
This represents the possible range of temperature-dependent error, scaled to temperature counts.

Then

(4) **Pcomp = Pcorr - 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.

Implementation: How to use:

On system startup:
Read the seven coefficients (A, B, C, D, E, TC50H, & TC50L) from sensor EEPROM, using the command sequence described in the datasheet section 'Digital Interface Command Formats'.

A, B, C & D are 32-bit signed integers, representing a scaled magnitude from -1.0 to +1.0.

E, TC50H, & TC50L are 8-bit signed integers, representing a scaled magnitude from -1.0 to +1.0.

Code Example:
Declarations:

```
// I2C Input, output buffers:
unsigned char inbuf[32] = {0}, outbuf[32] = {0};
// ----- DLLR Coefficients ------
float DLLR_A = 0.0, DLLR_B = 0.0, DLLR_C = 0.0, DLLR_D = 0.0;
float DLLR_E = 0.0, TC50H = 0.0, TCH50L = 0.0;
int32_t i32A = 0, i32B =0, i32C =0, i32D=0,
int8_t i8E = 0, i8TC50H = 0, i8TC50L = 0;
int8_t success = 0;                    // I2C communication status
// default sensor I2C address:
uint8_t ui8Address = 0x29;
```

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

Extended Compensation Instructions (Cont'd)

After sensor power-on, read EEPROM coefficient values, convert signed 32-bit integers to Float:

```
outbuf[0] = 47;                                     // Address of A high word
success = DUT_I2C_Write(ui8Address, outbuf, 1);     // 1-byte request
Wait_ms(20);                                        // EEPROM access time
// Read returns [Status][MSB][LSB]:
success = DUT_I2C_Read(ui8Address, inbuf, 3);       // EEPROM result
i32A = (inbuf[1] << 24) | (inbuf[2] <<16);          // Assemble MSBs
outbuf[0] = 48;                                     // Address of A low word
success = DUT_I2C_Write(ui8Address, outbuf, 1);     // 1-byte request
Wait_ms(20);                                        // EEPROM access time
success = DUT_I2C_Read(ui8Address, inbuf, 3);       // EEPROM result
i32A |= ((inbuf[1] << 8) | (inbuf[2]));             // LSBs, for int32
// convert to float, normalized to +/- 1.0:
DLLR_A = ((float)(i32A))/((float)(0x7FFFFFFF));
```

Repeat the EEPROM reading process, for B, C, D:

```
outbuf[0] = 49;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32B = (inbuf[1] << 24) | (inbuf[2] <<16);
outbuf[0] = 50;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32B |= ((inbuf[1] << 8) | (inbuf[2]));
DLLR_B = (float)(i32B)/(float)(0x7FFFFFFF);

outbuf[0] = 51;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32C = (inbuf[1] << 24) | (inbuf[2] <<16);
outbuf[0] = 52;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32C |= ((inbuf[1] << 8) | (inbuf[2]));
DLLR_C = (float)(i32C)/(float)(0x7FFFFFFF);

outbuf[0] = 53;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32D = (inbuf[1] << 24) | (inbuf[2] <<16);
outbuf[0] = 54;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i32D |= ((inbuf[1] << 8) | (inbuf[2]));
DLLR_D = (float)(i32D)/(float)(0x7FFFFFFF);
```

Now read the 8-bit coefficients TC50H, TC50L, and E, converting signed 8-bit integers to float:

```
outbuf[0] = 55;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i8TC50H = inbuf [1];                          // High Byte,
i8TC50L = inbuf [2];                          // Low Byte
// convert to float, normalized to +/- 1.0:

TC50H = (float)(i8TC50H)/(float)(0x7F);
TC50L = (float)(i8TC50L)/(float)(0x7F);

outbuf[0] = 56;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i8E = inbuf [2];                              // Low Byte
DLLR_E = (float)(i8E)/(float)/(0x7F);
```

These floating-point values are used in adjustments applied to each reading, described as follows:

Correction applied to each reading:

For each pressure value read from the sensor (POut), calculate

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

(If the application operates in an environment consistently near 25C, TCadj is an optional term.)

Example:

Start sensor reading:

```
outbuf[0] = 0xAD;                        // Request 4x oversampled reading
success = DUT_I2C_Write(ui8Address, outbuf, 1) // send 1-byte request
```

After conversion delay (or on rising EOC pin), read result:

```
success = DUT_I2C_Read(ui8Address, inbuf, 7); // read 7 bytes: Status, P, T
```

The sensor 24-bit pressure and temperature values are now in the I2C input buffer, preceded by the status byte. That is, inbuf[0] = Status Byte, inbuf[1], [2], [3] contain the pressure value, and inbuf[4], [5],[6] contain temperature.

Now, apply the corrective adjustments:

Declarations & definitions:

```
// constants:
const int32_t Tref_Counts = 8724019; // temperature counts at 25C
const float TCKScale = 100.0 * 100.0 * 134218.0 // scale TC50 to 0.6% FS0

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) 0x7FFFFF;
```

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

```
AP3 = DLLR_A * Pnorm * Pnorm * Pnorm;    // A*Pout^3
BP2 = DLLR_B * Pnorm * Pnorm;            // B*Pout^2
CP = DLLR_C * Pnorm;                     // C*POut
Corr = AP3 + BP2 + CP + DLLR_D;          // Linearity correction term
Pcorr = Pnorm + Corr;                    // Corrected P, range +/-1.0.
```

At this point, Pcorr represents the linearity-corrected pressure, optimized at 25C temperature. For room-temperature applications, this may be sufficient compensation.

Converting back to the sensor native format of unsigned 24-bit integer:

```
iPcorr = (int32_t)(Pcorr * (float)0x7FFFFF);    // Convert to signed 23-bit
iPcorr += 0x800000;                             // Back to unsigned 24-bit
```

The Pressure Output Transfer Function on Page 4 can then be applied, using iPcorr as the value for Pdig.

For systems exposed to temperatures significantly above or below 25C, additional temperature compensation may be applied, as shown below:

```
// 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 = TC50H;
else
       TC50 = 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;
```

Now, the temperature-dependent adjustment as a function of pressure and temperature difference from reference values as in Equation (3):

```
Tcorr = (1.0 - (DLLR_E * 2.5 * Pdiff)) * Tdiff * TC50 / TCKScale;
```

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

```
PCorrt = Pnfso - Tcorr; // corrected P: float, [0 to +1.0)
```

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

```
Pcomp = (uint32_t) (PCorrt * (float)0xFFFFFF);
```

Start next sensor reading:

```
outbuf[0] = 0xAD;                          // Request 4x oversampled reading
success = DUT_I2C_Write(ui8Address, outbuf, 1) // send 1-byte request
```

Extended Compensation Instructions (Cont'd)

Convert to pressure units:

The **Pcomp** *result represents the corrected output of the sensor, to be used in the Pressure Output Transfer Function as Pdig when computing pressure in calibrated measurement units.*

For illustration, if the sensor has a differential calibrated range of +/- 10 inH2O,

$$P_{inH2O} = 1.25 * ((Pcomp - (0.5 * 2^{24}))/ 2^{24}) * 2 * 10 \text{ inH2O}$$

where the 1.25 factor represents the scaling of full-scale output to the calibrated range (Output at Minimum pressure = 10% of full scale, output at Maximum pressure = 90%); and division of (Pcomp – 2^{23}) by 2^{24} resulting in a +/-0.5 scaling value re-scaled by the units 2x multiplier.

If a reading from this sensor results in Pcomp = 9145890 counts,

$$P_{inH2O} = 1.25 * ((9145890 – 8388608)/ 16777216) * 20 \text{ inH2O}$$
$$= 1.1284 \text{ inH2O}$$

For a compensated reading of Pcomp = 3000000 counts,

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

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

How to Order

Refer to Table 8 for configuring a standard base part number which includes the pressure range, package and temperature range. Table 9 shows the available configuring options. The option identifier is required to complete the device part number. Refer to Table 10 for the available device packages.

Example P/N with options: DLLR-L10D-E1NS-C-NAV6

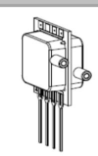
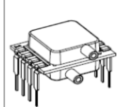
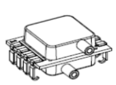
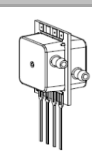
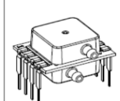
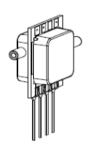
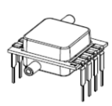
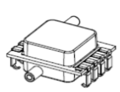
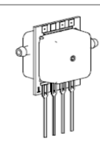
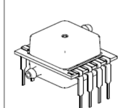
Table 8 - How to configure a base part number

	SERIES	PRESSURE RANGE			PACKAGE									TEMPERATURE RANGE	
					Base	Port Orientation		Lid Style		Lead Type					
ORDERING INFORMATION	ID	ID	Description		ID	ID	Description	ID	Description	ID	Description			ID	Description
	DLLR	L10D	±10 inH2O		E	1	Dual Port Same Side	N	Non-Barbed	S	SIP (see note 11)			C	Commercial
		L10G	0 to 10 inH2O			2	Dual Port Opposite Side	B	Barbed	D	DIP				
		L30D	±30 inH2O							J	J-Lead SMT				
		L30G	0 to 30 inH2O												
Example	DLLR -	L10D		-	E	1		N		S		-		C	

Table 9 - How to configure an option identifier

	COATING		INTERFACE		SUPPLY VOLTAGE		RESOLUTION	
ORDERING INFORMATION	ID	Description	ID	Description	ID	Description	ID	Description
	N	No Coating	A	Auto I2C, address 0x29/SPI	V	1.68V to 3.6V	6	16 Bit
			2	Auto I2C, address 0x28/SPI			7	17 bit
			3	Auto I2C, address 0x38/SPI			8	18 bit
			4	Auto I2C, address 0x48/SPI				
			5	Auto I2C, address 0x58/SPI				
			6	Auto I2C, address 0x68/SPI				
			7	Auto I2C, address 0x78/SPI				
Example	N		A		V		6	

Table 10 - Available E-Series Package Configurations

Port Orientation	Non-Barbed Lid				Barbed Lid			
	Lead Style				Lead Style			
	SIP (1)	DIP	J Lead SMT	Low Profile DIP	SIP (1)	DIP	J Lead SMT	Low Profile DIP
Dual Port Same Side	 E1NS	 E1ND	 E1NJ	N/A	 E1BS	 E1BD	N/A	N/A
Dual Port Opposite Side	 E2NS	 E2ND	 E2NJ	N/A	 E2BS	 E2BD	N/A	N/A
Single Port (Gage)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

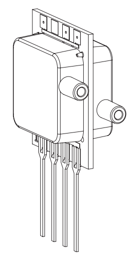
Specification Notes (Cont.)

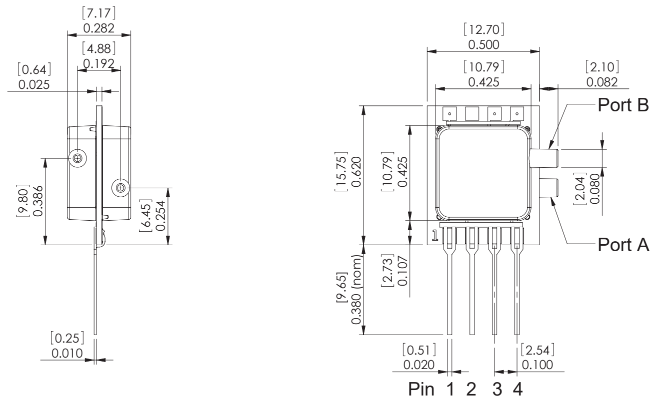
NOTE 11: SPI INTERFACE IS ONLY AVAILABLE IN 8-LEAD DIP OR J-LEAD PACKAGES.

Package Drawings

E1NS Package

Pinout
1) Gnd
2) Vs
3) SDA
4) SCL

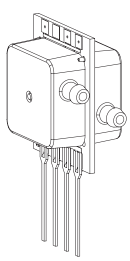


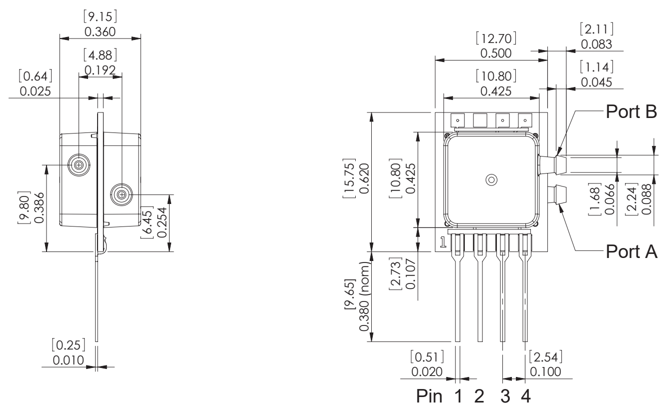


NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

E1BS Package

Pinout
1) Gnd
2) Vs
3) SDA
4) SCL





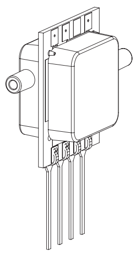
NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

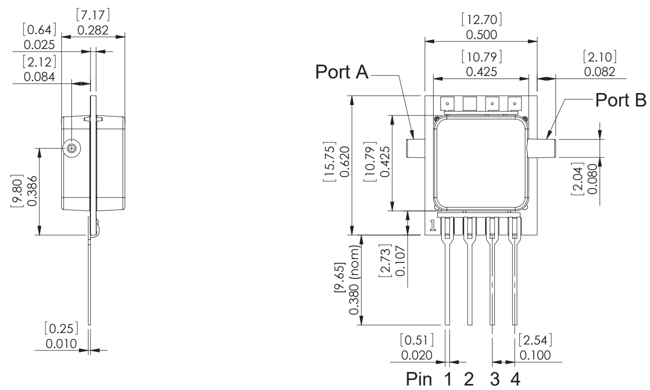
ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

Package Drawings (Cont'd)

E2NS Package

Pinout
1) Gnd
2) Vs
3) SDA
4) SCL

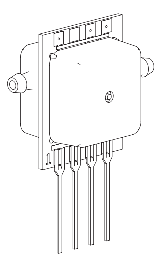


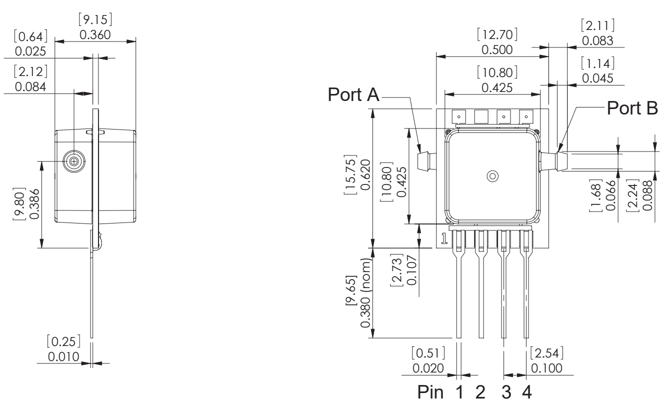


NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

E2BS Package

Pinout
1) Gnd
2) Vs
3) SDA
4) SCL

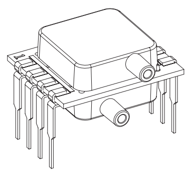


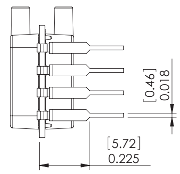


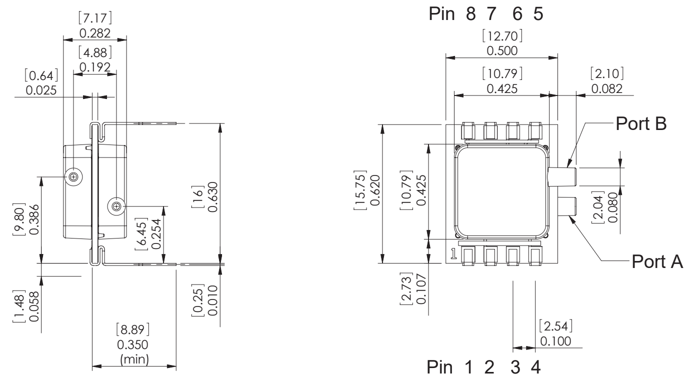
NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

E1ND Package

Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS



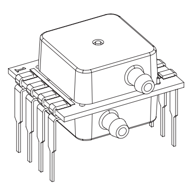


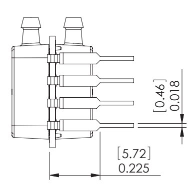


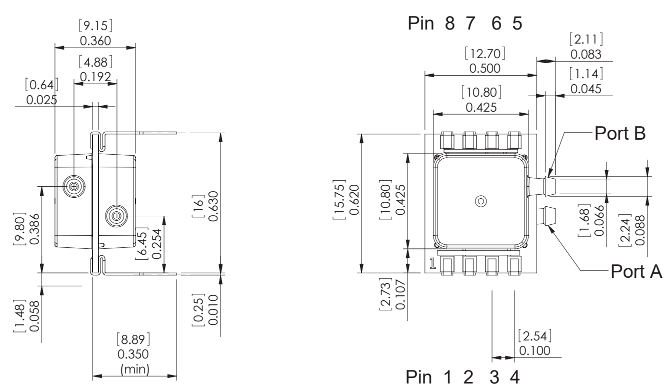
NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

E1BD Package

Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS





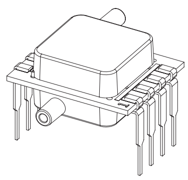


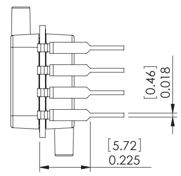
NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

ALL SENSORS
E www.allsensors.com
F 408 225 2079
P 408 225 4314
A 16035 Vineyard Blvd. Morgan Hill, CA 95037

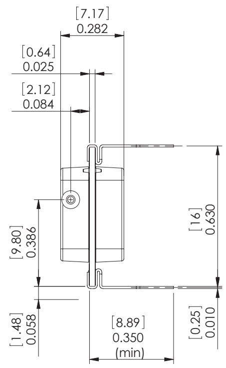
Package Drawings (Cont'd)

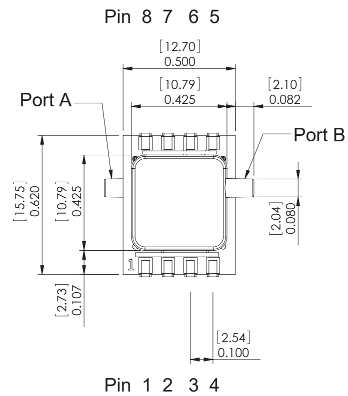
E2ND Package





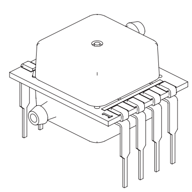
Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS

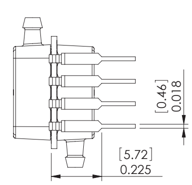




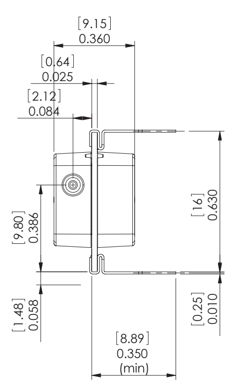
NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

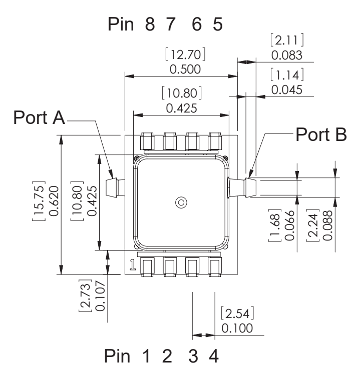
E2BD Package





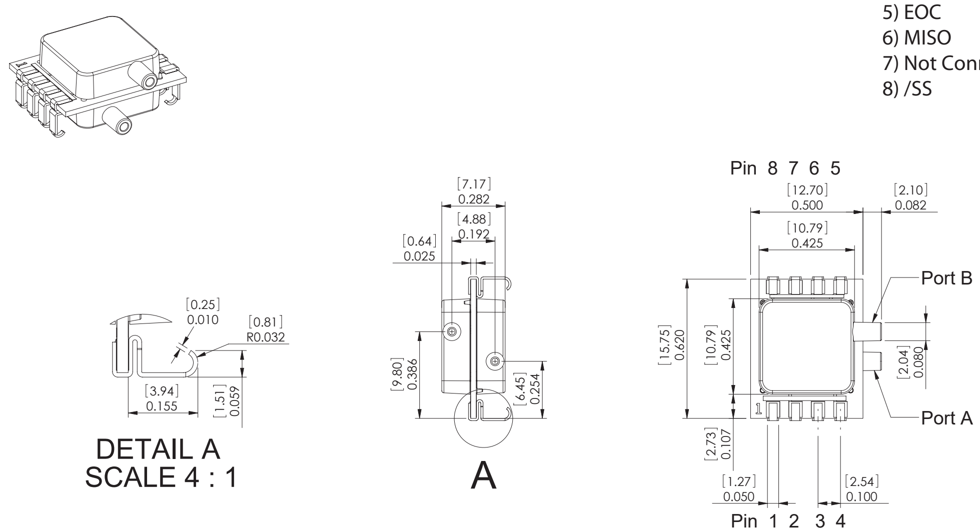
Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS





NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

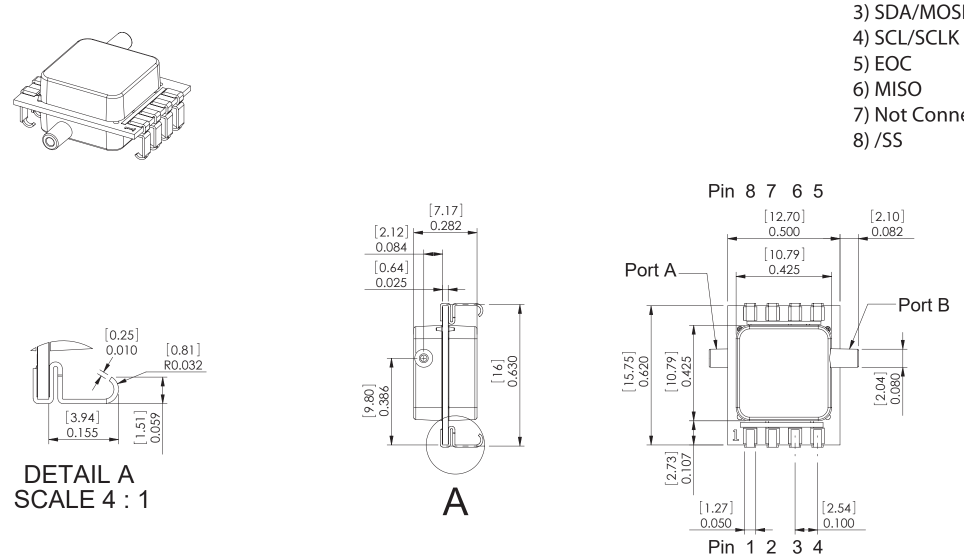
E1NJ Package



Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS

NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-10

E2NJ Package

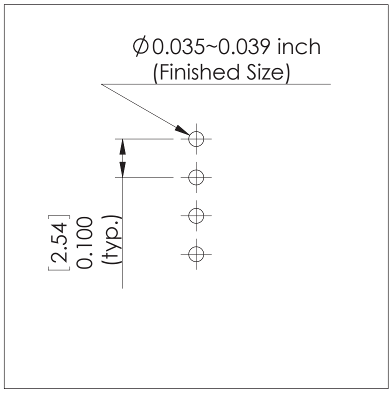


Pinout
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS

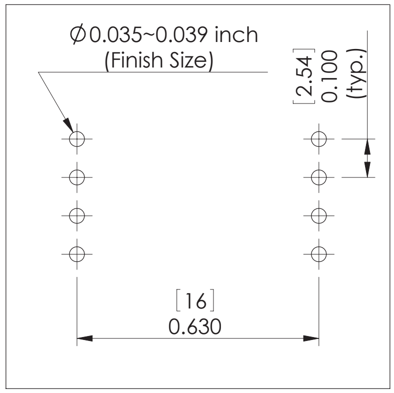
NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-10

ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037 P 408 225 4314 F 408 225 2079 E www.allsensors.com

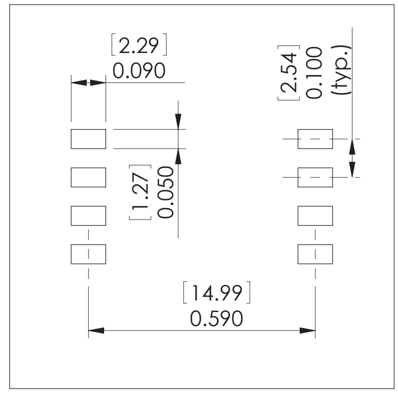
Suggested Pad Layout



PAD-01

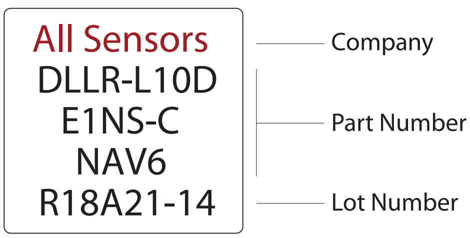


PAD-03



PAD-10

Product Labeling



Example Device Label