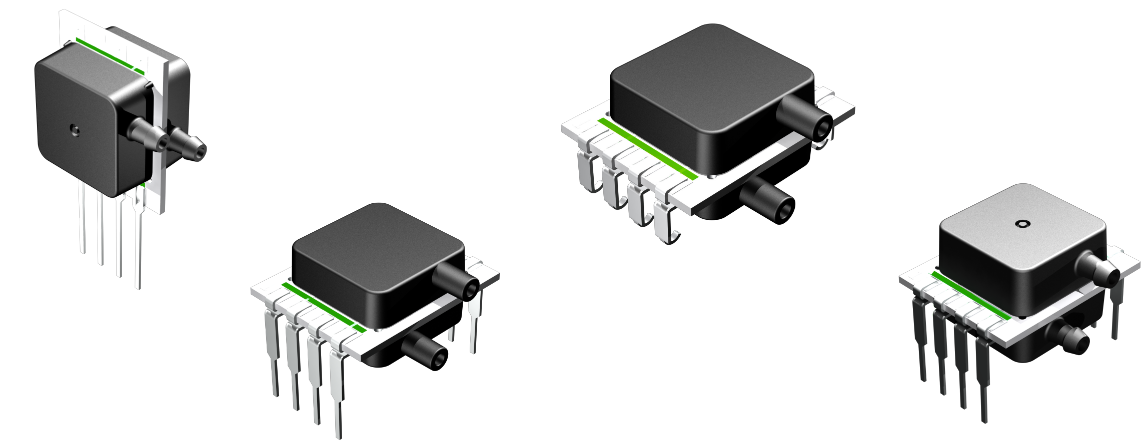
# ALL SENSORS®

## 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.

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

## Features

- 10 & 30 inH2O Pressure Ranges
- 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

## 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 | ±0.4 * $2^{24}$ |
| DLLR-L10G | 0 to 10 | 2488.4 | 100 | 25 | 300 | 75 | 0.8 * $2^{24}$ |
| DLLR-L30D | ± 30 | 7465.2 | 100 | 25 | 300 | 75 | ±0.4 * $2^{24}$ |
| DLLR-L30G | 0 to 30 | 7465.2 | 100 | 25 | 300 | 75 | 0.8 * $2^{24}$ |

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

## Pressure Sensor Maximum Ratings

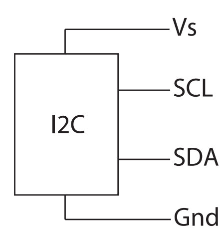
| | |
|---|---|
| **Supply Voltage (Vs)** | 3.63 Vdc |
| **Common Mode Pressure** | 10 psig |
| **Lead Temperature (soldering 2-4 sec.)** | 270 °C |

## Environmental Specifications

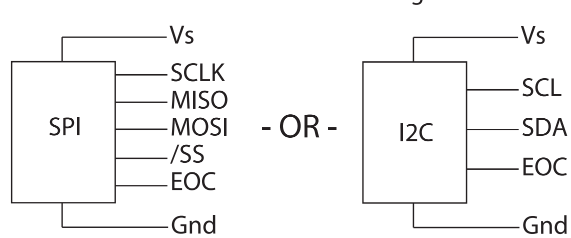
| | | |
|---|---|---|
| **Temperature Ranges** | | |
| Compensated: | Commercial | 0°C to 70°C |
| Operating | | -25°C to 85 °C |
| Storage | | -40°C to 125 °C |
| **Humidity Limits (non condensing)** | | 0 to 95% RH |

## Electrical Block Diagram

For SIP Packages




For DIP and J-Lead Packages




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



**Table of Contents**

## 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 | Minimum | Typical | Maximum | Units | Specification 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 |

## I2C / SPI Electrical Parameters for DLLR

| Parameter | Symbol | Min | Typ | Max | Units | Notes |
|---|---|---|---|---|---|---|
| Input High Level | - | 80.0 | - | 100 | % of Vs | 5 |
| Input Low Level | - | 0 | - | 20.0 | % of Vs | 5 |
| Output Low Level | - | - | - | 10.0 | % 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) | $C_{I2C\_IN}$ | - | - | 10.0 | pF | 5 |

## Pressure Output Transfer Function

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

Where:

$Pout_{dig}$ 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

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

Where:

$Tout_{dig}$ The sensor 24-bit digital temperature output.
(Note that only the upper 16 bits are significant)

**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.

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



Table of Contents

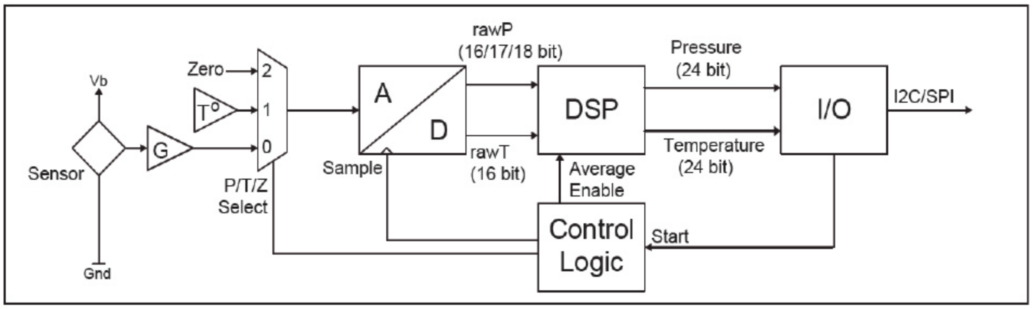
## Device Ordering Options

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.

## 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.

<u>Sensor Commands:</u> 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.*

<u>I/O Interface Configuration:</u> 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.*

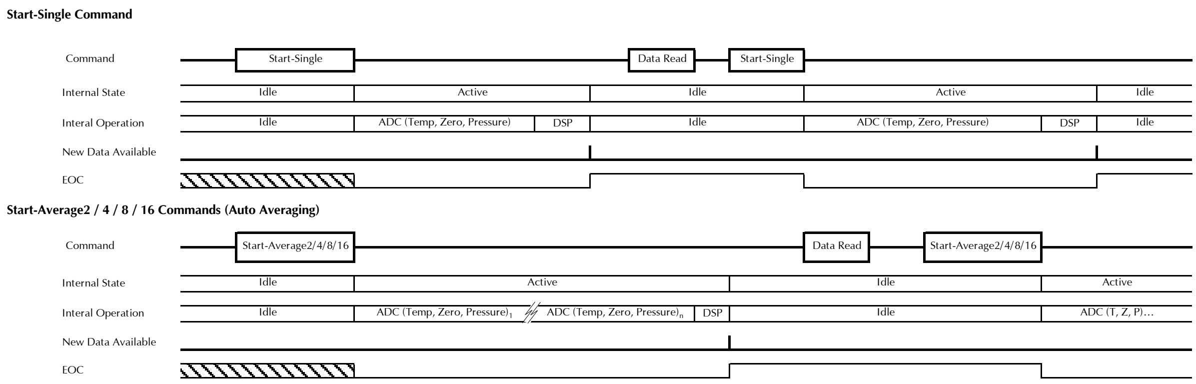
ALL SENSORS
A 16035 Vineyard Blvd. Morgan Hill, CA 95037   P 408 225 4314   F 408 225 2079   E www.allsensors.com



Table of Contents

## Operation Overview

### Figure 2 - DLLR Communication




## 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.**

### 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<br>***Host must send [0xF0], then 6 bytes of [0x00] on MOSI***<br>*Sensor Returns 7 bytes on MISO* |

| Read Sensor Status | |
|---|---|
| I2C | Read of 1 byte from device. |
| SPI | Read of 1 byte from device<br>***Host must send [0xF0] on MOSI***<br>*Sensor Returns 1 byte on MISO* |




**Table of Contents**

## Digital Interface Command Formats

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 High 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]<br>[TC50L] | [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 high 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><br>(*Values 47 -56 only* ) | 0x00 | 0x00 | <EEPROM Address><br>(*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.<br>***Host <u>must</u> send [0xF0], then 2 bytes of [0x00] on MOSI.***<br>*Sensor returns 3 bytes on MISO.* |

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

Write <**0x31**> , and read back: <Status> <**BHW**>.
Write <**0x32**>, and read back: <Status> <**BLW**>.
**B** = [BHW:BLW], assembling BHW and BLW into a signed 32-bit integer.

*Example : SPI Read of Coefficient **D** :*

Write <**0x35**><0x00><0x00>,
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>,
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.

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



**Table of Contents**

## 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. Refer to *'Extended Compensation Instructions Section'* for improving the accuracy of output pressure values.

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 overall data format of the sensor. Table 6 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 | Mode: [00 = Normal Operation ] |
| 2 | Memory Error [ 1 = EEPROM Checksum Fail] |
| 1 | Sensor Configuration [ always = 0] |
| Bit 0 [LSB] | ALU Error [1 = Error] |

## I2C Interface

### I2C Command Sequence

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, 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.**



Table of Contents

## 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.

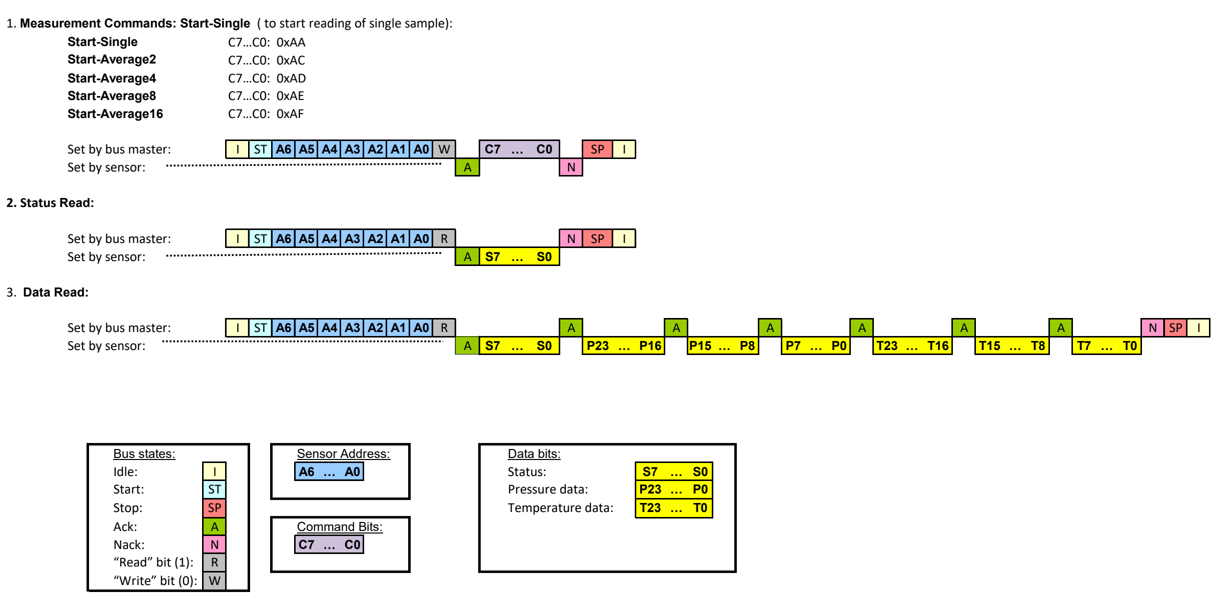
Slave address (An): The I2C-bus requires a unique address for each device. The DLLR sensor has a preconfigured slave address (defined by device option, see Table 9). After setting a START condition the master sends the address byte containing the 7 bit sensor address followed by a data direction bit (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




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



Table of Contents

## 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**

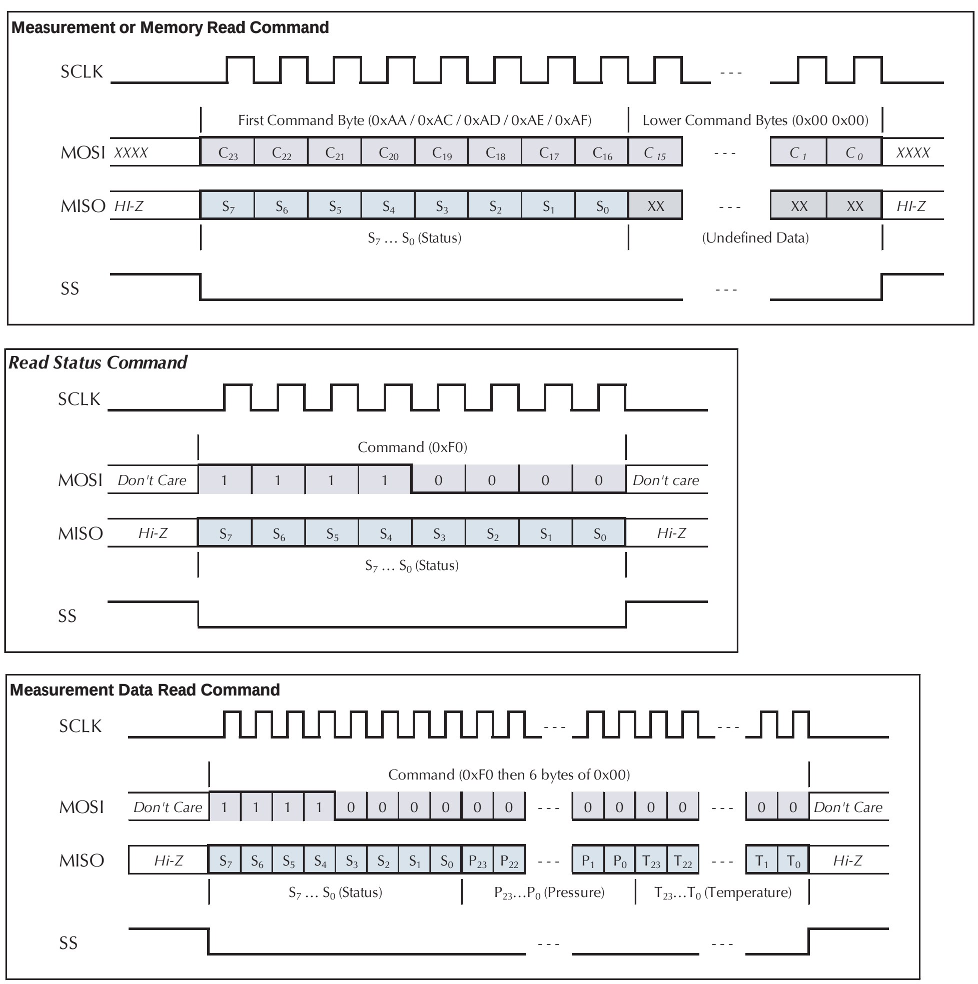
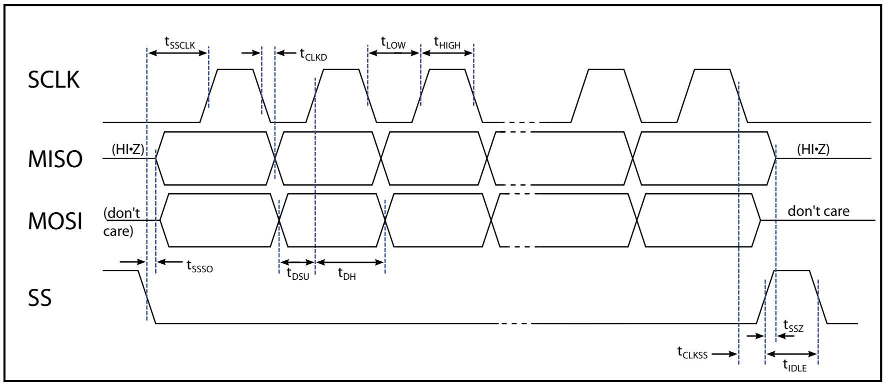





**Table of Contents**

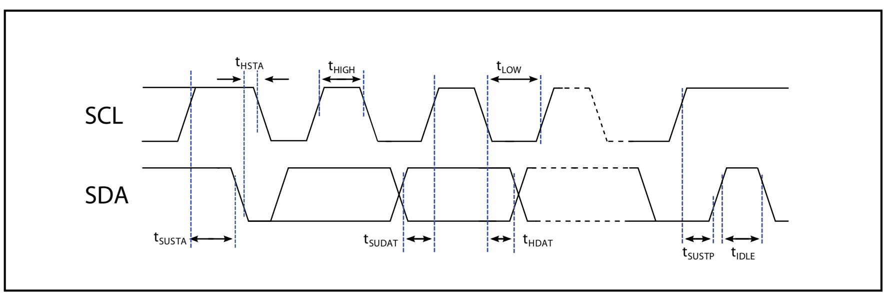
Figure 5 - SPI Timing Diagram




| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| SCLK frequency (1) | fSCLK | 0.05 | - | 5 | MHz |
| SS low to first clock edge | tSSCLK | 120 | - | - | ns |
| SS low to serial out | tSSSO | -- | - | 20 | ns |
| Clock to data out | tCLKD | 8 | - | 32 | ns |
| SCLK low width | tLOW | 100 | - | - | ns |
| SCLK high width | tHIGH | 100 | - | - | ns |
| Data setup to clock | tDSU | 50 | - | - | ns |
| Data hold after clock | tDH | 50 | - | - | ns |
| Last clock to rising SS | tCLKSS | 0 | - | - | ns |
| SS high to output hi-Z | tSSZ | -- | - | 20 | ns |
| Bus idle time | tIDLE | 250 | - | - | ns |

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

Figure 6 - I2C Timing Diagram




| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| SCL frequency | fSCL | 100 | - | 400 | KHz |
| SCL low width | tLOW | 1.3 | - | - | us |
| SCL high width | tHIGH | 0.6 | - | - | us |
| Start condition setup | tSUSTA | 0.6 | - | - | us |
| Start condition hold | tHSTA | 0.6 | - | - | us |
| Data setup to clock | tSUDAT | 0.1 | - | - | us |
| Data hold to clock | tHDAT | 0 | - | - | us |
| Stop condition setup | tSUSTP | 0.6 | - | - | us |
| Bus idle time | tIDLE | 2.0 | - | - | us |

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



Table of Contents

## 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.
The four linearity coefficients are obtained for each sensor at the factory by a 3rd order minimization solution to

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

For improved accuracy over temperature, residual temperature dependent errors are minimized by the term:
TCadj = (1 - (E * 1.25 * | 0.5 - P |)) * (T - Tref) * TC50
where:
TC50 = TC50H for T > Tref
TC50 = TC50L for T ≤ Tref

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.

***Example:***

```
// 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;
```

After sensor power-on:

```
outbuf[0] = 47;                                   // Address of A high word
success = DUT_I2C_Write(ui8Address, outbuf, 1); //  1-byte request
Wait_ms(20);  // EEPROM access time : 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]));      // assemble LSBs, for int32
DLLR_A = ((float)(i32A))/((float)(0x7FFFFFFF));  // convert to float

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);


outbuf[0] = 55;
success = DUT_I2C_Write(ui8Address, outbuf, 1);
Wait_ms(20);
success = DUT_I2C_Read(ui8Address, inbuf, 3);
i16E = ((inbuf[1] << 8) | (inbuf[2]));
i8TC50H = inbuf [1]; i8TC50L = inbuf [2];
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 [1];
DLLR_E = (float)(i8E)/(float)/(0x7F);
```

Correction applied to each reading:
For each pressure value read from the sensor (POut), calculate

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

***Example:***

(Start first reading:)

```
outbuf[0] = 0xAD;                  // Avg4 request = 0xAD
 rc = DUT_I2C_Write(ui8Address, outbuf, 1) // send 1-byte request
```

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 conversion delay (or on EOC pin), read result and apply correction:

```
rc = DUT_I2C_Read(ui8Address, inbuf, 7); // read 7 bytes: Status, P, T
float AP3, BP2, CP, LCorr, PCorr, Padj, TCadj, TC50;
int32_t iPraw, Tdiff, Tref, iTemp, iPCorrected;
uint32_t uiPCorrected;

//DLLR: Modify sensor P value:
iPraw = (inbuf[1]<<16) + (inbuf[2]<<8) + inbuf[3] - 0x800000;
iTemp = (inbuf[4]<<16) + (inbuf[5]<<8) + inbuf[6];
Pnorm = (float)iPraw;
Pnorm /= (float) 0x7FFFFF;
AP3 = DLLR_A * Pnorm * Pnorm * Pnorm;          // A*POut2
BP2 = DLLR_B * Pnorm * Pnorm;                  // B*POut2
CP = DLLR_C * Pnorm;                           // C*POut
LCorr = AP3 + BP2 + CP + DLLR_D;               // Linearity correction term

// Compute Temperature - Dependent Adjustment:
Tref = (int32_t)((2^24)*65/125); // Reference Temperature, in sensor counts
Tdiff = iTemp - Tref;
//TC50: Select High/Low, based on sensor temperature reading:
if (iTemp > Tref)
       TC50 = TC50H;
else
       TC50 = TC50L;
if (Pnorm > 0.5)
       Padj = Pnorm - 0.5;
else
       Padj = 0.5 - Pnorm;
TCadj = (1.0 - (DLLR_E * 1.25 * Padj)) * Tdiff * TC50;

PCorr = Pnorm + LCorr + TCadj;                 // corrected P: float, ±1.0
iPCorrected = (int32_t)(PCorr*(float)0x7FFFFF); // corrected P: signed int32

//corrected P: 24-bit unsigned value same unsigned format as sensor output

uiPCorrected = (uint32_t) (iPCorrected + 0x800000);
```

(Start next reading:)

```
outbuf[0] = 0xAD;                              // Avg4 request = 0xAD
rc = DUT_I2C_Write(ui8Address, outbuf, 1)      // send 1-byte request
```

**Convert to pressure units:**
The **PCorrected** result represents the corrected, signed 24-bit output of the sensor (**iPCorrected** in example code).
This dimensionless value is then used to compute the final result in appropriate units.
For example, if the calibrated range is +/- 10 $inH_2O$,

$$P_{inH2O} = 1.25 * (Pcorrected / 2^{23}) * 10\ inH_2O$$

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 by $2^{23}$ resolves Pcorrected ( range +/- $2^{23}$ ) into a +/-1.0 scaling value.

## 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

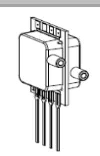
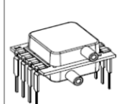
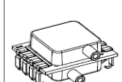
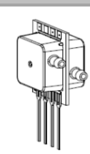
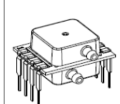
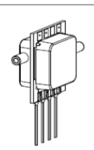
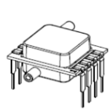
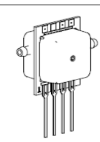
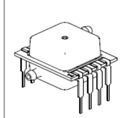
Table 8 - How to configure a base part number

| ORDERING INFORMATION | SERIES | PRESSURE RANGE | | | PACKAGE | | | | | | | TEMPERATURE RANGE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Base | Port Orientation | | Lid Style | | Lead Type | | | |
| | 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

| ORDERING INFORMATION | COATING | | INTERFACE | | SUPPLY VOLTAGE | | RESOLUTION | |
|---|---|---|---|---|---|---|---|---|
| | 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 PACKAGES.

ALL SENSORS

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

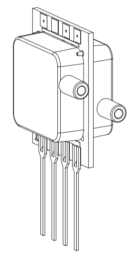


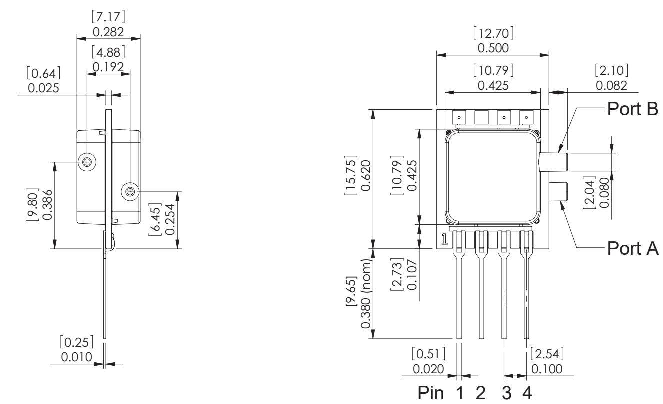
Table of Contents

## Package Drawings

### E1NS Package

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

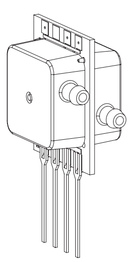


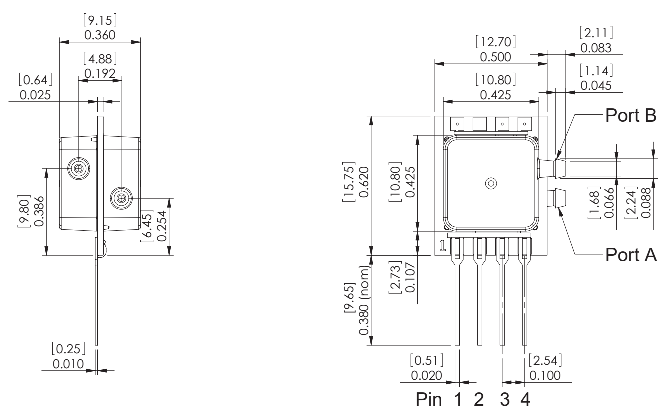



NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

### 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

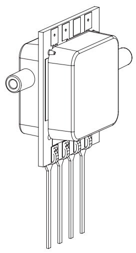


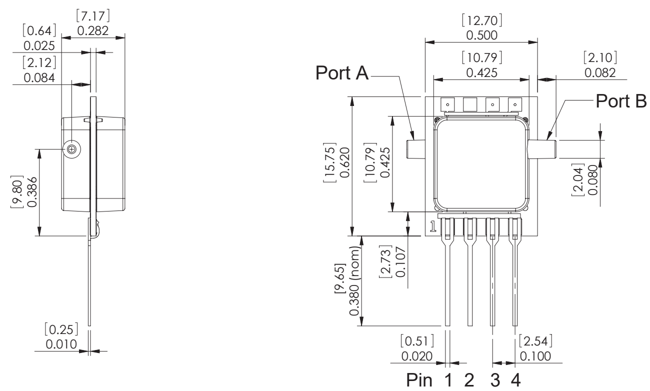
Table of Contents

## Package Drawings (Cont'd)

### E2NS Package

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

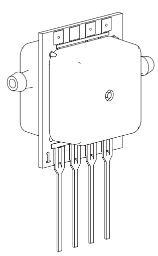


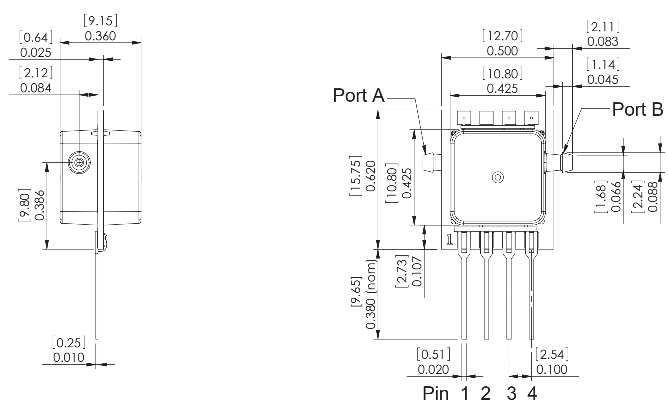



NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

### E2BS Package

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



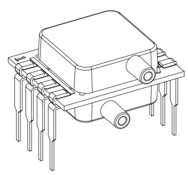


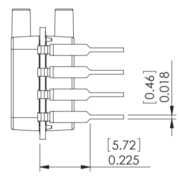

NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-01

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

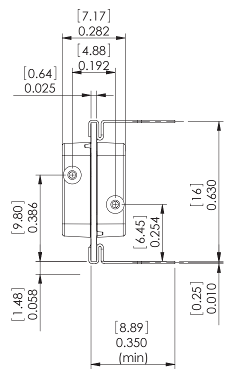
## Package Drawings (Cont'd)

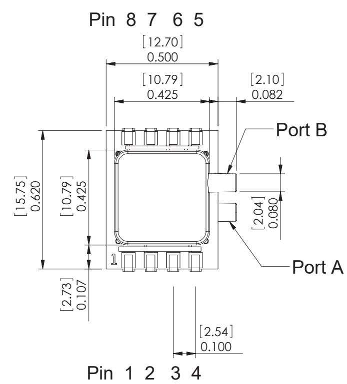
### E1ND Package





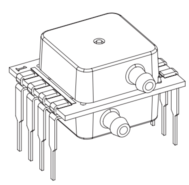

**Pinout**
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS

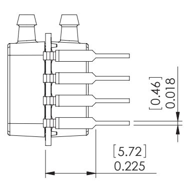


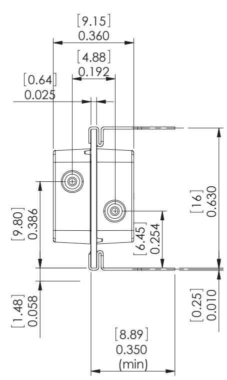

NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

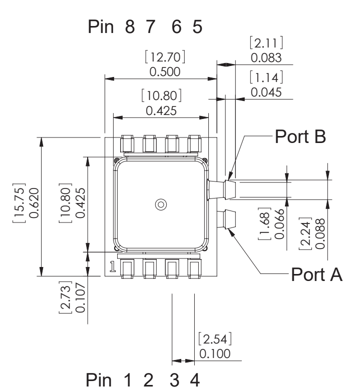
### E1BD Package






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



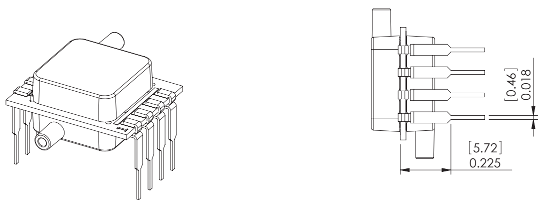




NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

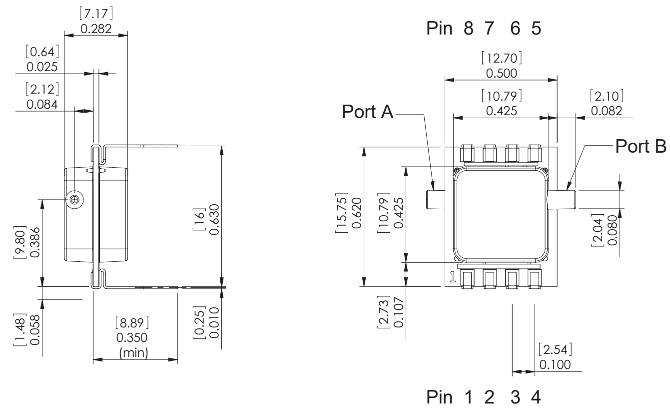


Table of Contents

## E2ND Package



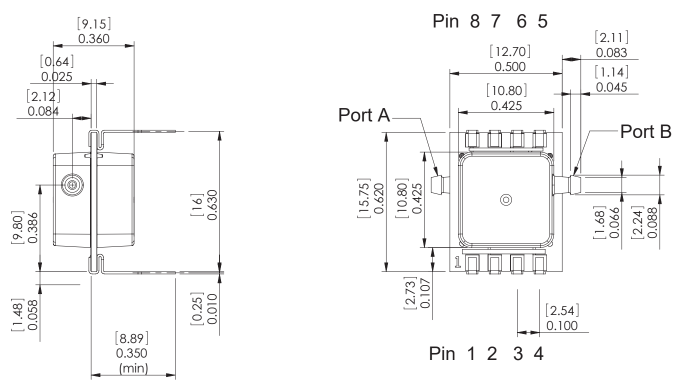

**Pinout**
1) Gnd
2) Vs
3) SDA/MOSI
4) SCL/SCLK
5) EOC
6) MISO
7) Not Connected
8) /SS




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

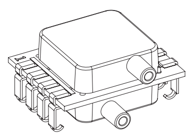



NOTES
1) Dimensions are in inches [mm]
2) For suggested pad layout, see drawing: PAD-03

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

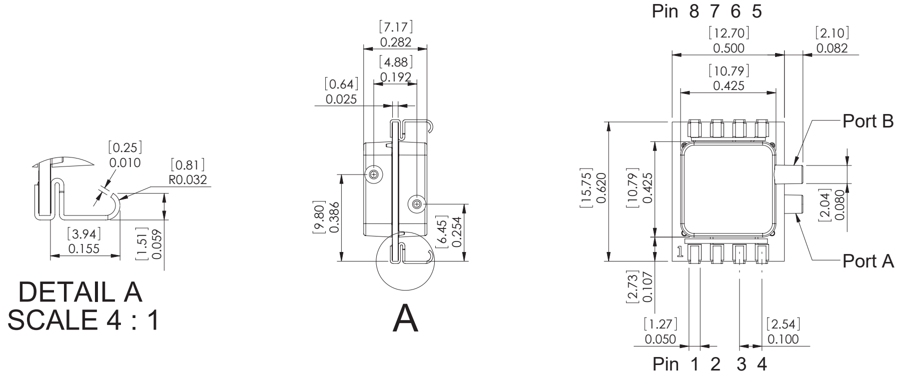
## Package Drawings (Cont'd)

### E1NJ Package



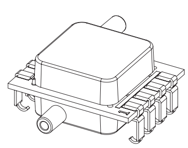
**Pinout**

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



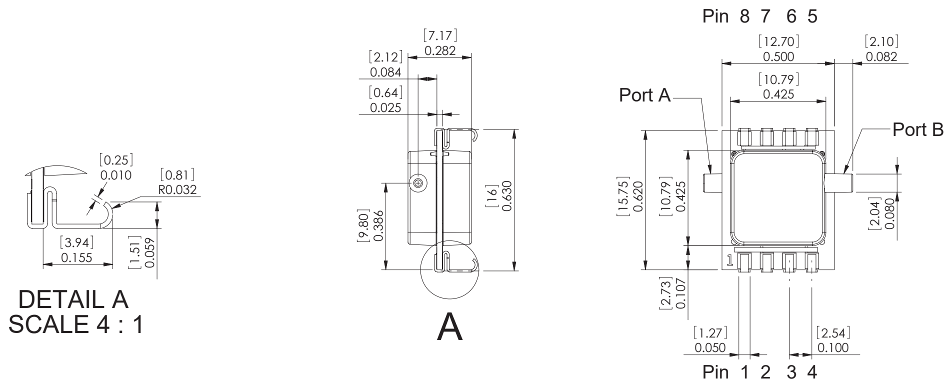

NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-10

### E2NJ Package



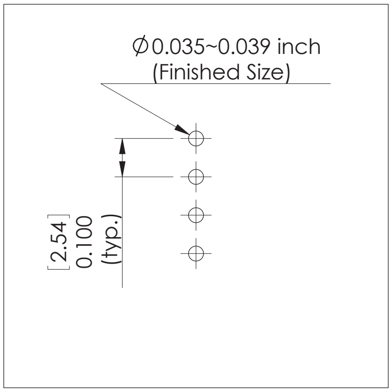
**Pinout**

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

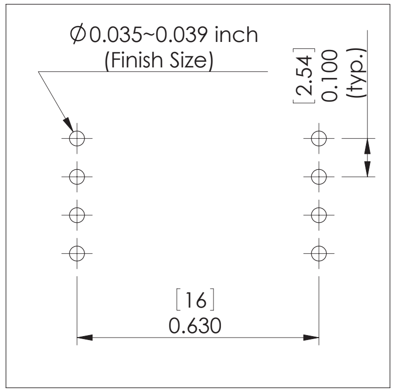



NOTES
1)Dimensions are in inches [mm]
2)For suggested pad layout, see drawing: PAD-10

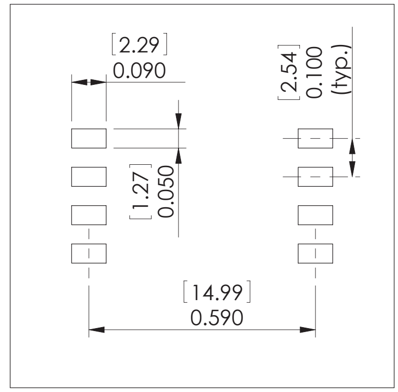
## Suggested Pad Layout




**PAD-01**

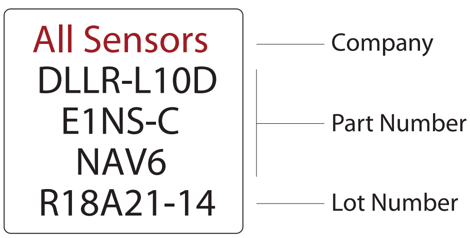



**PAD-03**




**PAD-10**

## Product Labeling




Example Device Label



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



Table of Contents

Живое партнерство

ООО "ЛайфЭлектроникс" "LifeElectronics" LLC

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

*Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.*

*С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров*

*Мы предлагаем:*

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

*В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.*

*Конструкторский отдел помогает осуществить:*

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

*Тел: +7 (812) 336 43 04 (многоканальный)*
*Email: org@lifeelectronics.ru*

***www.lifeelectronics.ru***