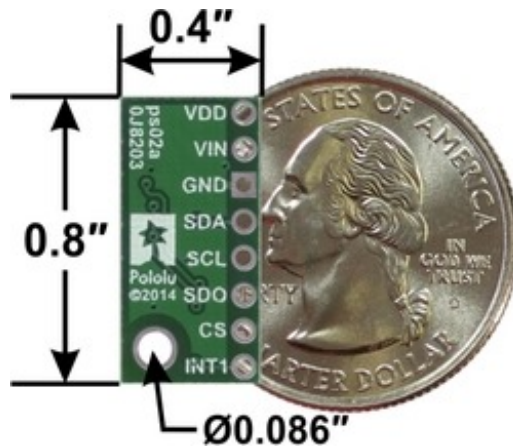


## LPS25H Pressure/Altitude Sensor Carrier with Voltage Regulator



### Overview

This board is a compact (0.4" × 0.8") carrier for ST's LPS25H MEMS absolute pressure sensor, or barometer; we therefore recommend careful reading of the LPS25H datasheet (1MB pdf) before using this product. The LPS25H is a great IC, but its small, leadless, LGA package makes it difficult for the typical student or hobbyist to use. It also operates at voltages below 3.6 V, which can make interfacing difficult for microcontrollers operating at 5 V. This carrier board addresses these issues by incorporating additional electronics, including a 3.3 V voltage regulator and level-shifting circuits, while keeping the overall size as compact as possible. The board ships fully populated with its SMD components, including the LPS25H, as shown in the product picture.

Compared to the earlier LPS331AP, the LPS25H features improved accuracy and reduced noise in the output. The addition of a built-in FIFO (First In, First Out) buffer allows the sensor to store pressure readings for burst transmission, reducing overall power consumption by allowing the host processor to sleep longer between data requests. Alternatively, the FIFO can be configured to perform a running average of pressure readings to further decrease the output noise.

This LPS25H carrier board is pin-compatible with our LPS331AP carrier, but due to the removal of one of the interrupt pins on the LPS25H, the position of the mounting hole has changed relative to the remaining pins. The two sensors use the same I<sup>2</sup>C addresses, but some of their configuration register fields are different, so code written to interface with an LPS331 might need to be modified to work with an LPS25H.

The LPS25H features embedded temperature compensation and has many configurable options, including selectable resolutions, a choice of output data rates, several FIFO operating modes, and a programmable external interrupt signal. Its pressure output has an absolute accuracy as low as ±0.2 mbar (0.02 kPa), with RMS noise of 0.01 mbar (0.001 kPa) in the highest-resolution mode with embedded filtering enabled. Pressure and temperature sensor data are available through a digital interface, which can be configured to operate in either I<sup>2</sup>C or SPI mode, and can be used for altimetry. (See the Sample Code section below for an Arduino library that can be used to turn this sensor into an altimeter.)

The carrier board includes a low-dropout linear voltage regulator that provides the 3.3 V required by the LPS25H, which allows the sensor to be powered from a 2.5 V to 5.5 V supply. The regulator output is available on the VDD pin and can supply almost 150 mA to external devices. The breakout board also includes a circuit that shifts the I<sup>2</sup>C/SPI clock and data in lines to the same logic voltage level as the supplied VIN, making it simple to interface the board with 5 V systems, and the board's 0.1" pin spacing makes it easy to use with standard solderless breadboards and

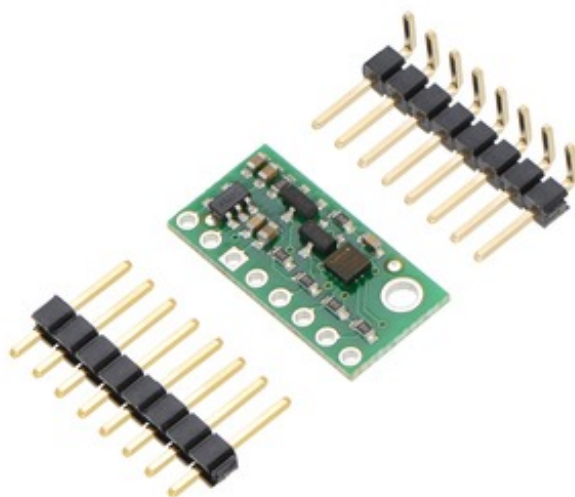
0.1" perfboards.

### Specifications

- **Dimensions:** 0.4" × 0.8" × 0.1" (10 mm × 20 mm × 3 mm)
- **Weight without header pins:** 0.5 g (0.02 oz)
- **Operating voltage:** 2.5 V to 5.5 V
- **Supply current:** 2 mA
- **Output format (I<sup>2</sup>C/SPI):** 24-bit pressure reading (4096 LSb/mbar)
- **Sensitivity range:** 260 mbar to 1260 mbar (26 kPa to 126 kPa)

### Included components

A 1×8 strip of 0.1" header pins and a 1×8 strip of 0.1" right-angle header pins are included, as shown in the picture below. You can solder the header strip of your choice to the board for use with custom cables or solderless breadboards, or you can solder wires directly to the board itself for more compact installations.



The board has one mounting hole that works with #2 and M2 screws (not included).

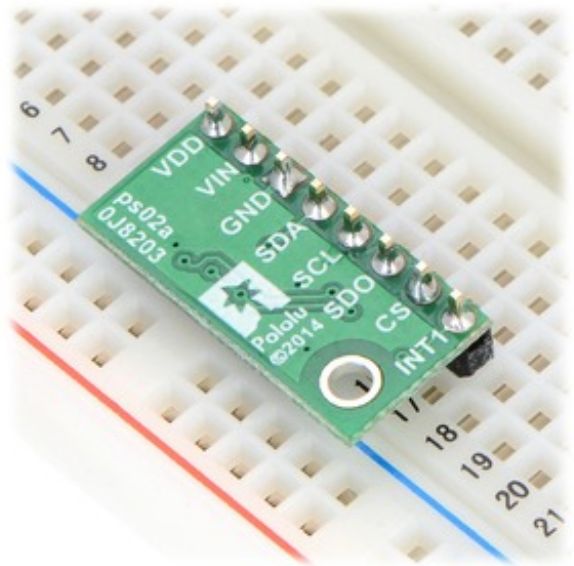
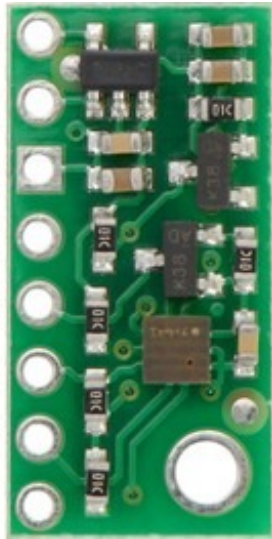
### Using the LPS25H Connections

Regardless of the interface being used to communicate with the LPS25H, its VIN pin should be connected to a 2.5 V to 5.5 V source, and GND should be connected to 0 volts. (Alternatively, if you are using the sensor with a 3.3 V system, you can leave VIN disconnected and bypass the built-in regulator by connecting 3.3 V directly to VDD.)

A minimum of two logic connections are necessary to use the LPS25H in I<sup>2</sup>C mode (this is the default mode): SCL and SDA. These pins are connected to built-in level-shifters that make them safe to use at voltages over 3.3 V; they should be connected to an I<sup>2</sup>C bus operating at the same logic level as VIN. The remaining pins are not connected to level-shifters on the board and are not 5V-tolerant, but our 4-channel bidirectional logic level shifter can be used externally with those pins to achieve the same effect.

To use the LPS25H in the default SPI mode, four logic connections are required: SPC, SDI, SDO, and CS. These should be connected to an SPI bus operating at the same logic level as VIN. The SPI interface operates in 4-wire mode by default, with SDI and SDO on separate pins, but it can be configured to use 3-wire mode so that SDO shares a pin with SDI.

**VDD (3.3V Out)**  
**VIN (2.5–5.5V)**  
**GND**  
**SDA/SDI/SDO**  
**SCL/SPC**  
**SDO/SA0**  
**CS**  
**INT1**



LPS25H pressure/altitude sensor carrier with voltage regulator, labeled top view.

LPS25H pressure/altitude sensor carrier with voltage regulator in a breadboard.

## Pinout

PIN	Description
VDD	Regulated 3.3 V output. Almost 150 mA is available to power external components. (If you want to bypass the internal regulator, you can instead use this pin as a 3.3 V input with VIN disconnected.)
VIN	This is the main 2.5 V to 5.5 V power supply connection. The SCL/SPC and SDA/SDI level shifters pull the I <sup>2</sup> C and SPI bus high bits up to this level.
GND	The ground (0 V) connection for your power supply. Your I <sup>2</sup> C or SPI control source must also share a common ground with this board.
SDA/SDI/SDO	Level-shifted I <sup>2</sup> C data line and SPI data in line (also doubles as SDO in 3-wire mode): HIGH is VIN, LOW is 0 V
SCL/SPC	Level-shifted I <sup>2</sup> C/SPI clock line: HIGH is VIN, LOW is 0 V
SDO/SA0	SPI data out line in 4-wire mode: HIGH is VDD, LOW is 0 V. This output is not level-shifted. Also used as an input to determine I <sup>2</sup> C slave address (see below).
CS	SPI enable (chip select). Pulled up to VDD to enable I <sup>2</sup> C communication by default; drive low to begin SPI communication.
INT1	Programmable interrupt, a 3.3-V-logic-level output. This output is not level-shifted.



enable SPI communication with microcontrollers operating at the same voltage as VIN (2.5 V to 5.5 V).

In the default 4-wire mode, the pressure sensor transmits data to the SPI master on a dedicated data out (SDO) line that is not level-shifted. If the SPI interface is configured to use 3-wire mode instead, the SDI line doubles as SDO and is driven by the LPS25H when it transmits data to the master. A detailed explanation of the SPI interface on the LPS25H can be found in its datasheet (1MB pdf).

### **Sample Code**

We have written a basic Arduino library for the LPS25H that makes it easy to interface this sensor with an Arduino. The library makes it simple to configure the LPS25H and read the raw pressure data through I<sup>2</sup>C, and it provides functions for calculating altitude based on the measured pressure for those looking to use this sensor as an altimeter.

### **Protocol hints**

The datasheet provides all the information you need to use this sensor, but picking out the important details can take some time. Here are some pointers for communicating with and configuring the LPS25H that we hope will get you up and running a little bit faster:

- **The pressure sensor is in power down mode by default. You have to turn it on by writing the appropriate value to the CTRL\_REG1 register to choose an output data rate.**
- **You can read or write multiple registers in a single I<sup>2</sup>C command by asserting the most significant bit of the register address to enable address auto-increment.**
- **You can enable the same auto-increment feature in SPI mode by asserting the second bit (bit 1, called the MS bit in the datasheet) of an SPI command.**

**[Documentation on producer website.](#)**