Air Velocity Digital Sensor

BM62S2301-1
Arduino Library V1.0.2 Description

Revision: V1.10 Date: August 19, 2024

www.bestmodulescorp.com

Contents

Introduction

The BM62S2301-1 is an air velocity digital sensor from Best Modules, which uses the I²C communication method. This document provides the description of the BM62S2301-1 Arduino Lib functions and how to install the Arduino Lib. The example uses the BME26M301 module to demonstrate the function of reading air velocity.

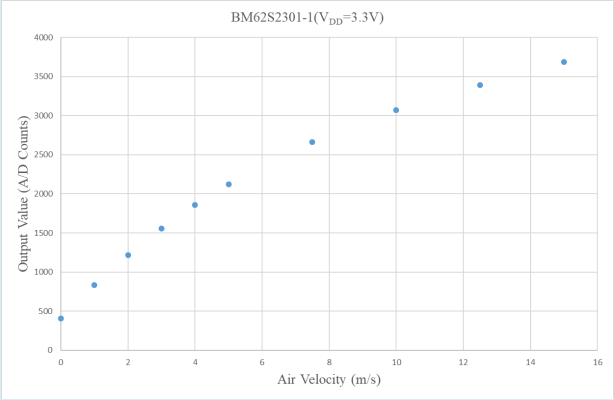
Applicable types:

Part No.	Description
BM62S2301-1	Air velocity digital sensor
BME26M301	On-board BM62S2301-1 sensor

Arduino Lib Functions

Arduino Lib Name: BM62S2301-1		Lib Version: V1.0.2
Constructors & Initialisation		
1	BM62S2301-1(TwoWire *theWire=&Wire, uint8_t iic_addr=BM62S2301_1_ADDRESS);	
	Description	Constructor
	Parameter	*theWire: I²C communication interface selection, defaults to Wire interface iic_addr: I²C slave address BM62S2301_1_ADDRESS: 0x28
	Return Value	—
	Note	—
2	void begin()	
	Description	Module initialisation
	Parameter	—
	Return Value	void
	Note	—
Performance Functions		
3	uint16_t readRawAirVelocity()	
	Description	Obtain the air velocity sensor digital A/D count value
	Parameter	—
	Return Value	Air velocity A/D Count value (12-bit)
	Note	The greater the A/D Count value the higher the air velocity; On the contrary, it indicates a smaller air velocity
4	float readAirVelocity(uint8_t uintChoose=1)	
	Description	Obtain the air velocity
	Parameter	uintChoose: Select the unit of returned air velocity 1: MPS(m/s) (default) 2: MPH
	Return Value	Air velocity value (unit: MPS or MPH)
	Note	According to the output curve (note) to convert the air velocity digital output A/D value into an air velocity value in m/s, or convert into an air velocity value in mph based on 1mph=2.236936×1m/s Refer to the BM62S2301-1 datasheet for details.

Note:



Air Velocity Value (m/s)	Output Value (A/D Count)
0	409
1	834
2	1215
3	1554
4	1856
5	2123
7.5	2664
10	3068
12.5	3391
15	3686

The BM62S2301-1 outputs the A/D Count values, the application terminal needs convert them to specific air velocity values (m/s) based on the output curve. When converting the air velocity values, take two adjacent points of the output curve as an interval, and approximate the curve as a linear output within the interval. After reading the air velocity A/D Count, find the interval corresponding to that A/D Count and convert it to obtain the specific air velocity value. The specific conversion method is as follows:

Assuming the read A/D Count value is 3000, 3000 is located within the interval between points (7.5, 2664) and (10, 3068)

Within this interval, the proportion of duty for 3000 is (3000-2664)/(3068-2664)≈83.2%

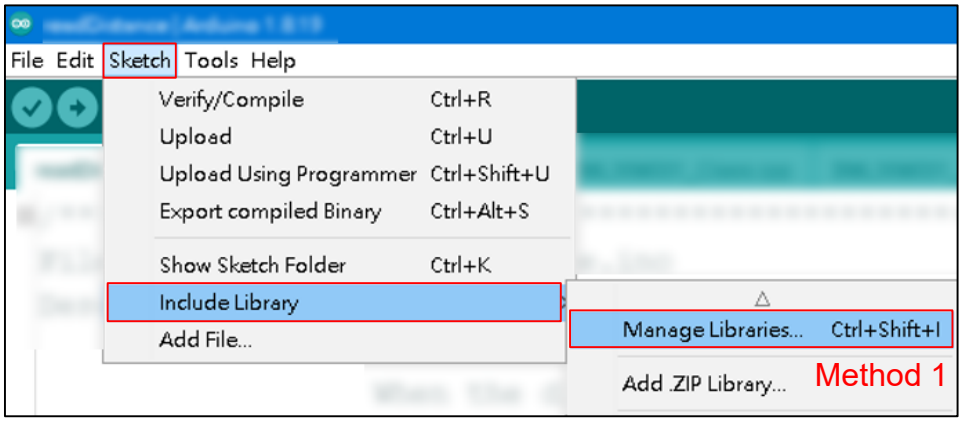
The specific air velocity value is: (10-7.5)×83.2%+7.5=9.58m/s

Arduino Lib Download and Installation

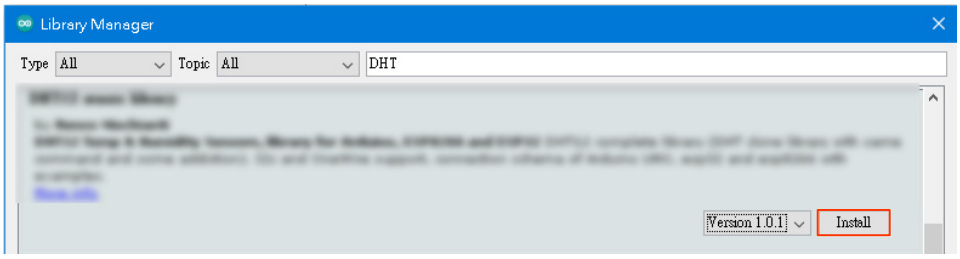
BM62S2301-1 Library: Refer to the following two methods to install the BM62S2301-1 Arduino Library

Method 1: Search for installation

Arduino IDE → Sketch→Include Library → Manage Libraries... → Search BM62S2301-1 → Install



Search for Installation Step 1



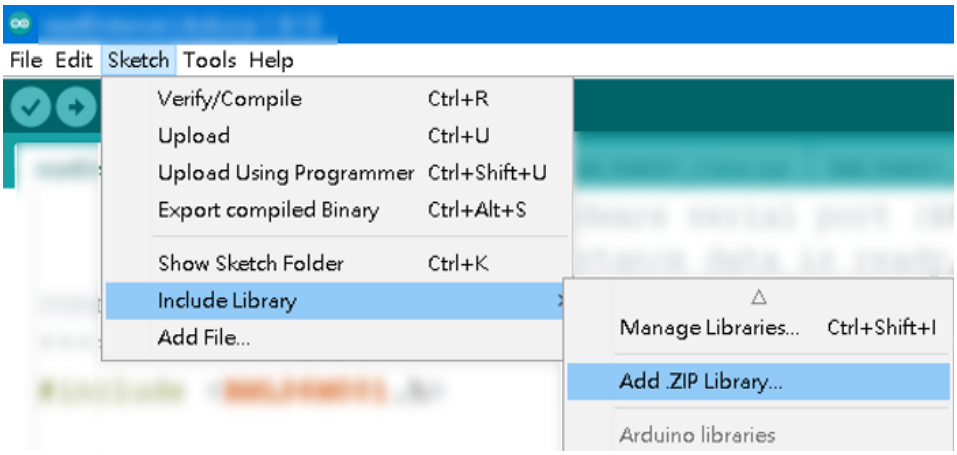
Search for Installation Step 2

Method 2: Download before adding a ZIP library

Download method: Open the Best Modules official website

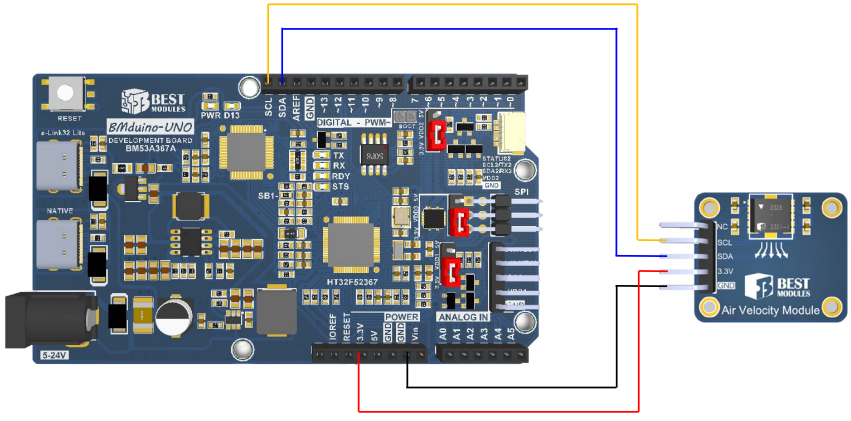
(https://www.bestmodulescorp.com/bm62s2301-1.html) and download the BM62S2301-1 Library from “Arduino example program” under the “DOCUMENTS” menu.

Add .ZIP library: Arduino IDE → Sketch → Include Library → Add .ZIP Library....



Arduino Example

Example: readAirVelocity



Physical Connection Diagram

Example function: Uses the I²C method to communicate with the module, obtain the air velocity value and display it in the serial monitor.

1. Open the example: File → Examples → Select Lib (BM62S2301-1) → Select example (readAirVelocity)
2. Example description:
   a. Create object & initialise object

```
#include "BM62S2301-1.h"
BM62S2301_1 AVS       // Create object
void setup() {
  AVS.begin();        // Initialise the module
  Serial.begin(9600); // Initialise the serial monitor with
                      // a baud rate of 96000
  /*Module status display*/
  Serial.println("Air Velocity Sensor Module");
  Serial.println("BM62S2301-1 has been powered on");
  Serial.println("==========================================");
}
```

   b. Obtain the air velocity data and display it in the serial monitor

```
void loop() {
  Serial.print("ADC count: ");
  Serial.print(AVS.readRawAirVelocity());
  Serial.print("\tm/s: ");
  Serial.print(AVS.readAirVelocity(1));
  Serial.print("\tmph: ");
  Serial.println(AVS.readAirVelocity(2));
  delay(500); // Set the serial to obtain the data every 500ms
}
```

3. Open the serial monitor and select the baud rate to be 9600. The serial monitor will display the data as follows: