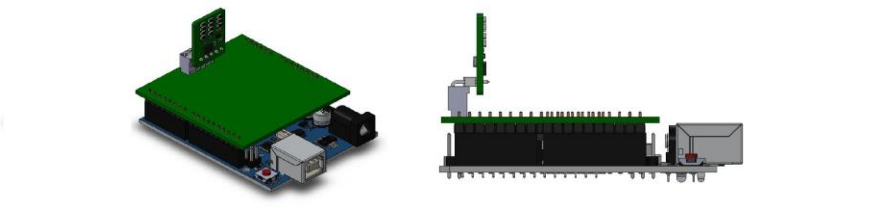
CALIBRATED INTRASENSE EVALUATION KIT

APPLICATION NOTE



Introduction

TE Connectivity's IntraSense is one of the smallest pressure sensors in the market and simplifies pressure monitoring directly in the anatomy, enabling more controlled procedures and better clinical data. The Calibrated IntraSense series are delivered with an attached PCBA. A 5-pin right angle header is soldered to the board for power supply to the circuit and to provide both digital and amplified analog noise-filtered outputs.

The IntraSense Evaluation Kit is intended for quick set-up and initial functional testing of the Calibrated IntraSense Product family. This document describes how to set up and use the evaluation kit based on an Arduino code and a LabVIEW SW application.

Description

The IntraSense series are absolute pressure sensors designed to fit into a 1-French hypo tube. The sensor comes pre-attached to cabling, simplifying the connection for the end user. This sensor compares pressure in vivo to an onboard vacuum cavity for reference to an absolute standard. It delivers accurate and stable pressure for acute procedures in the clinically useful range of -300 mmHg to +500 mmHg (460 mmHg to 1260 mmHg absolute) and from 10°C to 60°C.

The Calibrated IntraSense is a MEMS sensor (Distal End) attached to a 48 AWG trifilar cable for signal transmission (Figure 2) to a PCBA (Proximal End). The PCBA, (Figure 3), includes an ASIC to read and compensate both, temperature and pressure. The sensor has two outputs: a digital I2C and an amplified analog for external data processing.

a)

b)

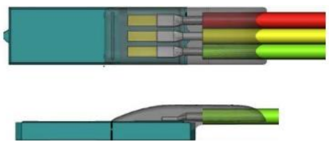
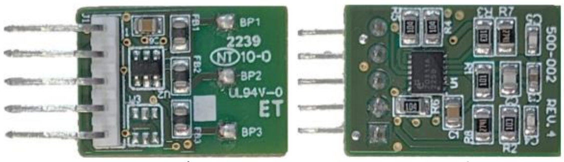


Figure 2. *Distal End views. a) Shows the top view of the MEMS. b) Is the right view of the Sensor.*



a) b)

Figure 3. *Proximal End views. a) Top view of the PCBA. b) Bottom view of the PCBA.*

IntraSense Evaluation Kit

The IntraSense Evaluation Kit is composed of:

- An Arduino UNO R3 Board preloaded with the TE code "IntraSense_Evalkit.ino" (Code is available for download at TE.com).
- USB A-B cable for PC-Arduino communication (Figure 16).
- A LabVIEW SW application "IntraSense Eval Kit VI" to display real-time charts of the sensed Temperature and Pressure, as well as the functionality to save the measured data. (Application SW is available for download at TE.com).
- The TE IntraSense *Arduino Shield* (Figure 5) for communication between the Calibrated IntraSense and the Arduino UNO.
- Five Calibrated IntraSense Sensors.

Arduino Code

The electronic design of the Arduino board allows the user to work directly with the microcontroller by programming in C++ language. The coding and programming interface is Arduino IDE SW (https://www.arduino.cc/en/software).

NOTE: This project was developed on Arduino Uno R3 board (Figure 4) and Arduino IDE version 2.2.1 SW.

Figure 4. *Arduino UNO R3 Board, top view.*

The IntraSense_Evalkit.ino code is shown below.

```
#include "Wire.h"
//ASIC REGISTERS
#define IntraSense 0x6C          //  I2C bus address
#define tRegister 0x2E           //  Address of temperature register
#define pRegister 0x30           //  Address of pressure register
#define sRegister 0x36           //  Address of status register

//PRESSURE CONSTANTS
#define PMin 460.0               //  IntraSense Minimum Pressure
#define min_dPout -26214.0       //  Digital value for 460 mmHg
#define PMax 1260.0              //  IntraSense Maximum Pressure
#define max_dPout 26214.0        //  Digital value for 1260 mmHg
#define dPspan 52428.0           //  Digital Pressure span=max_dPout-min_dPout
#define atmPressure 760.0        //  Atmospheric pressure at sea level

//STATUS CONSTANTS
int Wire_pwr = 128;       //Bit of PWR supply wire status from Status Register
int Wire_sig = 256;       //Bit of signal wire status from Status Register

//VARIABLES FOR TEMPERATURE/PRESSURES CALCULATION, AND STATUS CHECK
float Temp;
float Press;
float gagePressure;
int PWRWire;
int SIGNWire;

void setup() {
  delay(500);                    //  Setup guard
  Serial.begin(9600);            //  Baud rate
  Wire.begin();                  //  Function from Wire library
}

//SERIAL BUFFER PRINTING
void loop() {
  readIntraSense();
  Serial.println("_");
  Serial.print(Temp);            //  Temperature value printing
  Serial.print("_");
  Serial.print(Press);           //  Pressure value printing
  Serial.print("_");
  Serial.print(gagePressure);        // Approximated gage pressure printing
  Serial.print("_");
  Serial.print(PWRWire);             // Issue w/ pwr wire status printing
  Serial.print("_");
  Serial.print(SIGNWire);            // Issue w/ signal wires status printing
  delay(5000);                       // Sampling rate time (miliseconds)
}

//COMMUNICATION WITH THE SENSOR
void readIntraSense(){
  Wire.beginTransmission(IntraSense);     // Start communication with sensor
  Wire.write(tRegister);                  // Send start Address for read
  Wire.endTransmission(IntraSense);       // End the write

//READING OF THE TWO BYTES OF EACH REGISTER
  Wire.requestFrom(IntraSense, 6);        //  "Restart" bus and and read 6
                                          //   bytes from start address
  while (Wire.available()){               //  Check for data from slave
    int dTout = Wire.read();              //  read first Temperature byte
    dTout |= ((int)Wire.read()<<8);       //  read second Temperature byte
    int dPout = Wire.read();              //  read first Pressure byte
    dPout |= ((int)Wire.read()<<8);       //  read second Pressure byte
    int dStatus = Wire.read();            //  read first Status byte
    dStatus |= ((int)Wire.read()<<8);     //  read second Status byte

//CONVERTION OF DIGITAL PRESSURE COUNTS TO DECIMAL VALUES
    //Transfer function:
float totalPressure = PMin + ((dPout - min_dPout)/(dPspan))*(PMax-PMin);
    Press = totalPressure;
//CALCULATION OF GAGE PRESSURE
    gagePressure = totalPressure-atmPressure;
//TEMPERATURE SACALING FOR +/-16k for -40 to 125C
    Temp = (dTout * 0.0051562) + 42.5;
//LOGICAL 'AND' OPERATIONS TO KNOW IF THERE ARE WIRE COMMUNICATION ISSUES
    PWRWire = (dStatus & Wire_pwr);
    SIGNWire = (dStatus & Wire_sig);
  }
}
```

The code's function is to use the Arduino Board as I2C Master device to read the registers of temperature, pressure and status of the IntraSense's ASIC (the I2C slave in this case). Once the communication is established, the Arduino will send the sensed values to the LabVIEW Application. The code's structure is divided as follows:

1. The first part is the definition of the IntraSense's ASIC I2C address, as well as the temperature, pressure, and status registers. The information of the three registers is expressed in a signed 16-bits (2's complement) digital count value. For more information about the registers please refer to Application note 40AN7000 available at TE.com.
2. The next section is the definition of constants of the Calibrated IntraSense minimum and maximum values in decimal and their equivalent values in digital counts, the pressure span (or pressure range) and the atmospheric pressure value at sea level.
   a. 460 [mmHg] equivalent to -26,214 [counts].
   b. 1260 [mmHg] equivalent to 26,214 [counts].
   c. 52428 counts of span.
3. The Status Constants are bits to compare the Status Register information related to the wire.
4. The three *float* variables '**Temp'**, '**Press'**, and '**gagePressure'** are used to take and display the values of temperature and both absolute pressure and approximated gage pressure at sea level. The remaining two *int* variables '**PWRWire**' and '**SIGNWire**' are used to display the Status Register information and detect if there are issues related to the connection between ASIC and MEMS.
5. To display the data, the five variables are sent via serial communication in a constant loop. Every time the loop is initiated, a new sample of temperature and pressure is measured. For this project, the sampling rate time is 5 seconds (this value can be modified by the user by opening the IntraSense_Evalkit.ino code (line 59) in the Arduino IDE SW).
6. The communication between the Arduino UNO and the sensor starts with a request from the microcontroller to read the ASIC's temperature register. If the communication is successful, the next step is to read each register of interest.
7. The reading of the three registers (Temperature, Pressure and Status) is done one byte at the time, it means that six bytes (two from each register) will be read and contain the sensed value expressed in digital counts. The corresponding value of temperature, absolute and approximated gage pressure and information of status register will be saved in a provisional variable: ***dTout*** for Temperature, ***dPout*** for Pressure and ***dStatus*** for the status value. The first two variables will be mathematically transformed to a decimal value and sent to their corresponding global variable (Point 4). For the Status value, the digital count will be processed through two logical 'AND'

operators, one with an integer ‘256’ value and a second one with an integer ‘128’ value. In this way both wire status bits will be analyzed.

If LabVIEW is not available, the outcome values can be displayed in the Arduino IDE Serial Monitor as follows:

*_Temp_Press_gagePressure_PWRwire_SIGWire_*

IntraSense Evaluation Kit VI

The IntraSense Eval Kit VI is a SW application developed in NI LabVIEW 2023 Q1 (64-bits) version. For the correct functionality download and install the following packages from NI.com:

- **LabVIEW Runtime Engine**. This package will include the libraries needed to run the IntraSense Eval Kit VI. https://www.ni.com/es/support/downloads/software-products/download.labview-runtime.html#521521

- **NI VISA Driver**. This package includes the drivers to allow IntraSense Eval Kit VI to read the PC COM Ports. https://www.ni.com/es/support/downloads/drivers/download.ni-visa.html#521671

Besides both packages, an .aliases file and an .ini file are needed to run the ‘IntraSense Eval Kit VI’. These files are included in the same download as the ‘IntraSense Eval Kit VI’. **NOTE**: The three files must be located in the same folder/path for good function of the application (Figure 5).

Figure 5. *IntraSense Evaluation Kit VI files.*

After a double click on the IntraSense Eval Kit VI icon, the user window (Front panel) will be displayed (Figure 6).

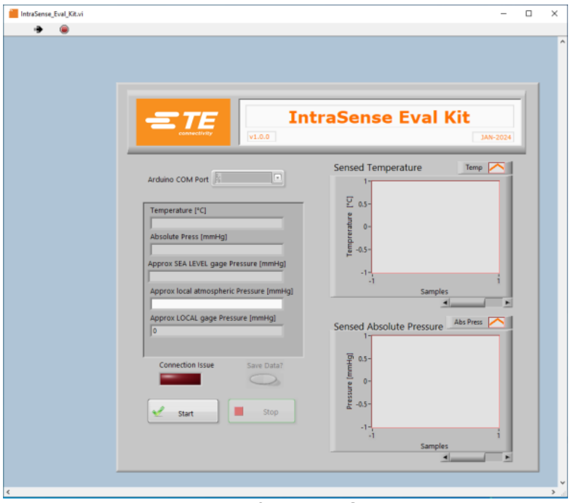


Figure 6. *Front Panel of the IntraSense Evaluation Kit VI.*

This front panel is composed of:

- A drop-down menu to select the Arduino UNO COM Port (Figure 7). To find the COM Port assigned by the PC to the user’s Arduino UNO the user must go to **Start** → **Settings** → **System** → **Device Manager**, or by typing “**Device Manager**” into the search bar. Once in Device Manager, look for the entry for “Ports (COM & LPT)” and expand it to view the available COM ports (Figure 8).

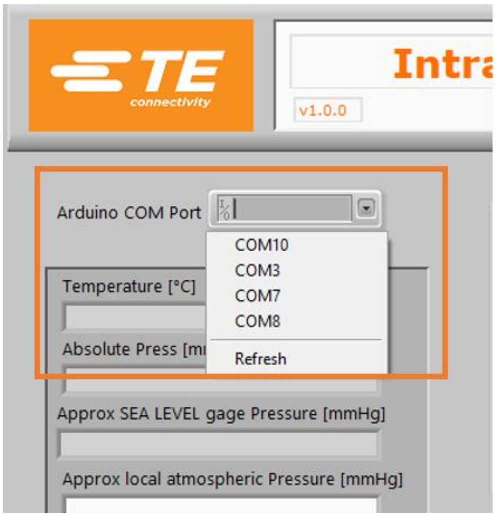


Figure 7. *COM Port menu of the IntraSense Evaluation Kit VI.*

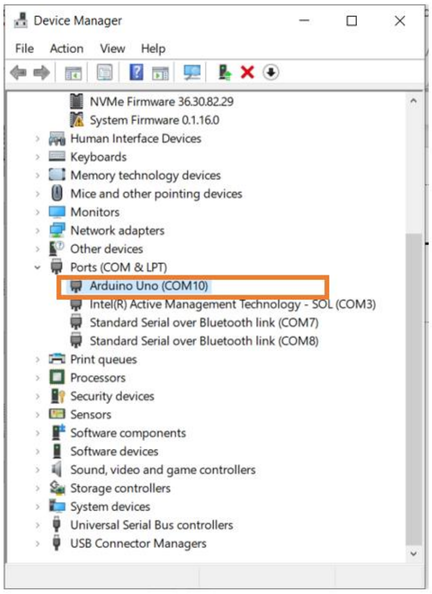


Figure 8. *Device Manager window. For this example, the COM Port assigned to the Arduino UNO is the COM 10.*

- Four indicators at the left side of the Front Panel to display the sensed Temperature, Absolute Pressure (actual pressure sensed by the MEMs), the Approximated Gage Pressure at Sea Level (subtraction of Absolute pressure minus 760 [mmHg]) and a last one to see the Approximated Gage Pressure at the local geographic point (subtraction of Absolute pressure minus the local pressure in mmHg). For the estimation of the local pressure, the user must write the local pressure value as shown in Figure 9. If the space remains blank, the application will display the same value as the Absolute Pressure. The temperature is calculated in Celsius degrees, and the units of all the pressures are millimeters of Mercury (mmHg).

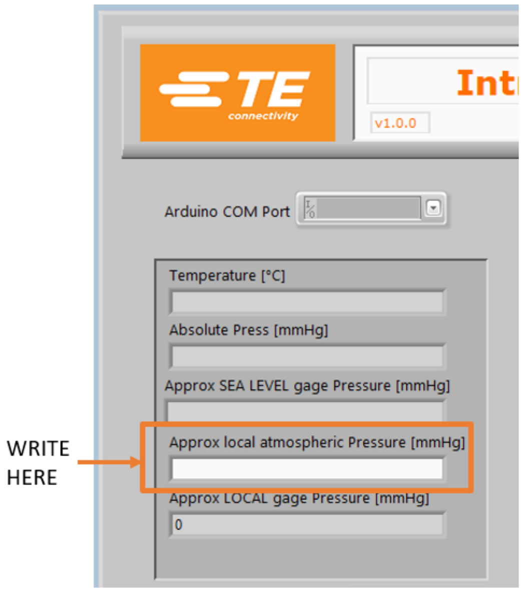


Figure 9. *Indicators of Temperature and Pressures*

- Two charts are placed on the right side of the Front Panel, one for Temperature and another for the Absolute Pressure (Figure 10). Both charts will update themselves by a new measure based on the sampling rate time defined in the IntraSense_Evalkit.ino code.

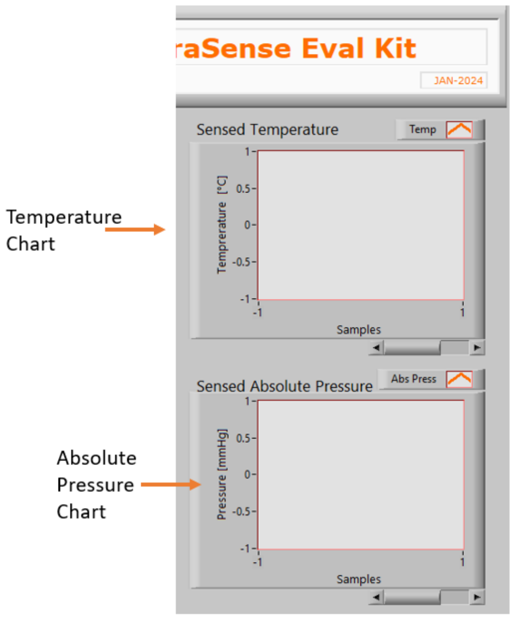


Figure 10. *Temperature and Absolute Pressure charts. The user can scroll each chart on the 'X' axis if needed.*

- Finally, a red LED and three buttons are located at the bottom of the left side of the Front Panel (Figure 11).
  - The 'Start' button initiates the SW application.
  - The 'Stop' button will stop the SW application.
  - The LED indicates a connection issue between the Distal End and the Proximal End.
  - The 'Save Data?' switch allows the user to save the sensed data (three pressure values and temperature) in an external CSV file. The switch will be enabled once the SW application starts to run.

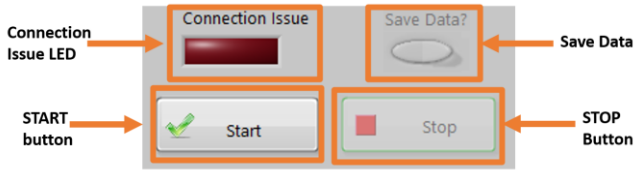


Figure 11. *Buttons of the Front Panel*

TE IntraSense *Arduino Shield*

The hardware interface to connect the Calibrated IntraSense with the Arduino Board is the TE IntraSense *Arduino Shield*, a printed circuit board that includes four male header connectors at the bottom to connect directly with the Arduino UNO and one female connector on the top to insert the Calibrated IntraSense Proximal End (Figure 12).

Each male connector is labeled to fit into the proper Arduino female connector.

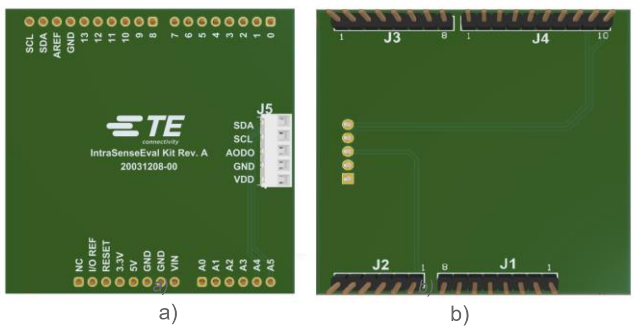


Figure 12. *Arduino Shield for Calibrated IntraSense Kit. a) Top layer of PCBA. b) Bottom layer of PCBA.*

All five connections for the male connector are indicated at the top layer, each signal (power, ground, digital and analog) are routed to the proper male header to connect with the Arduino UNO Board.

IntraSense PCBA (Proximal End)

A right-angle male header is soldered on the top side of the Calibrated IntraSense PCBA. This connector provides the connections to VDD and GND needed by the sensor, also is connected to both analog and Digital I2C outputs (Figure 13 for reference). For this Evaluation Kit, only the digital output is read by the Arduino Code.

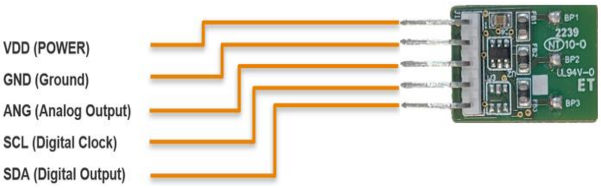


Figure 13. *Proximal End pin assignments.*

The Proximal End size is 21.08 mm x 14.08 mm, (not including pins). For more information, please see the Calibrated IntraSense datasheet 40DS1110.

Preparation

For shipping, the Calibrated IntraSense trifilar is spooled (Figure 14). The Distal End is secured with Kapton tape, and the Proximal End is held at the top of the spool and covered to protect the PCBA from damage caused by movement.

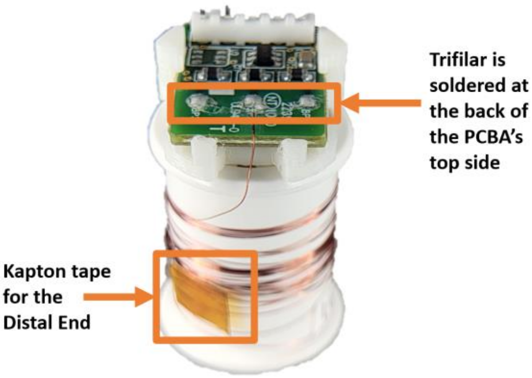
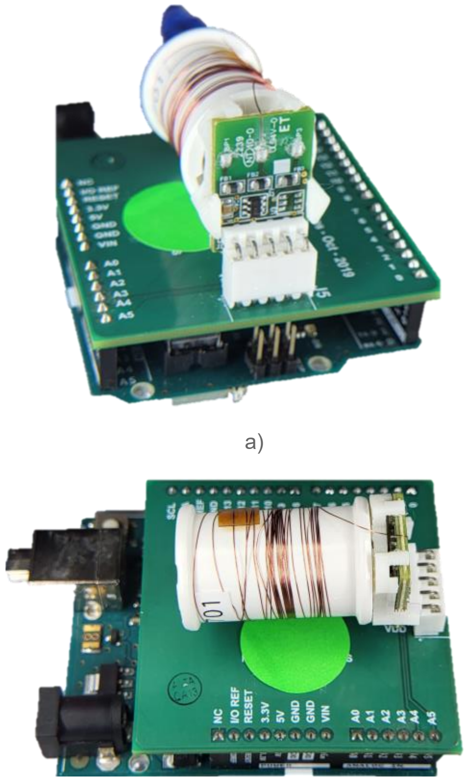


Figure 14. *Calibrated IntraSense back view with spooled wire and PCBA at the top of the spool.*

TE suggests the following procedure for de-spooling to minimize risk of breakage:

1. Prepare a clean, ESD-controlled workspace free from sharp objects or objects that could become entwined with the wire.
2. Remove the protection from the PCBA and then peel the tape off from the Distal End.
3. After the tape has been removed from the spool, unwind the wire gently. Unwind as much wire as needed to perform the measurement.
4. Connect the Proximal End to the IntraSense Evaluation Kit Arduino Shield. The correct placement of the Calibrated IntraSense PCBA with the Arduino Shield is shown in the Figure 15. (In this example the wire and the PCBA are not removed from the spool).
5. Use the USB A-B cable provided in the kit to connect the Arduino the PC. Connect the USB A male connector to the PC's USB port, and the USB B male connector to the Arduino.
6. At this point, all the hardware is ready to start the testing.



a)

b)

Figure 15. *Correct placement of sensor board into connector. a) Front View. b) Top-side view.*

a) b)

Figure 16. *a) USB A connector (PC). b) USB B connector (Arduino).*

Running the Calibrated IntraSense Eval Kit VI

Before opening the application, please ensure that all the hardware is set up correctly as explained in the previous section. Also, check at the 'Device Monitor' the assigned COM Port to your Arduino UNO (this step will confirm if the board is properly connected).

To initiate the testing, follow the steps below:

1. If the application is installed in the Desktop, click on the IntraSense Eval Kit Icon to open the application. The Calibrated IntraSense Eval Kit VI window will appear to the user.
2. Configure the COM Port for the Arduino connection and enter the Local atmospheric pressure.
3. Once step 2 is completed, click on 'Start' button to initiate the testing, if the selected COM Port is wrong, an error message will appear (Figure 17). Otherwise, the Arduino will start to send data.

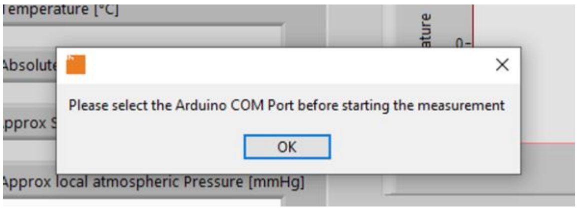


Figure 17. *Error message if the COM Port is wrong.*

During the data transmission, there are three possible conditions:

a. If no sensor is connected to the Arduino shield, the readings will be always 0 for the Temperature, Absolute Pressure and Approximated gage pressure at sea level, for the fourth reading, the approximated gage local pressure will be the same number than the user added but negative.

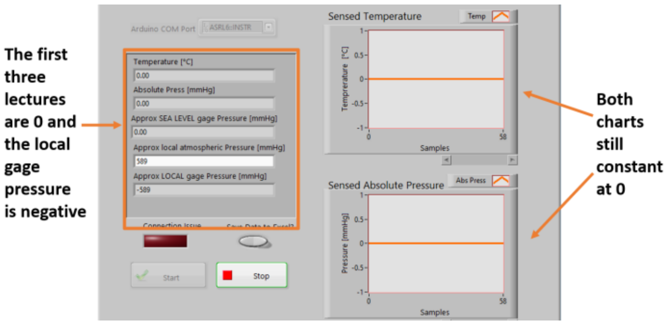


Figure 18. *Calibrated IntraSense KIT VI main window without sensors.*

b. The next case is when the connection between the Distal End and the Proximal End has an issue. If this is the case, the red LED will be turned on and the temperature indicator and chart will show an extreme high value (Figure 19). The reason of this condition may be physical damage in the trifilar (broken or twisted wires), or a bad soldering of the wires at the Distal End or at the PCBA.

TE suggests, for this case, to stop the application, disconnect the hardware and unplug the Distal End from the TE IntraSense *Arduino Shield*, in order to check the trifilar for a broken or twisted wire. Another option to verify the trifilar status is to use a multimeter to check the continuity between the bond pads of the PCBA (where the trifilar is soldered, Figure 14), if there is a short circuit or an open circuit between the bond pads, the trifilar may have physical damage.

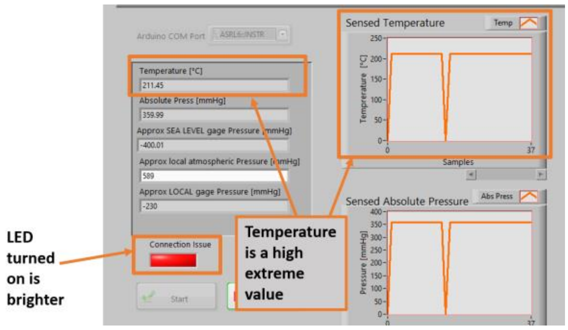


Figure 19. *Calibrated IntraSense KIT VI main window with wire issues.*

c. The third case is when the sensor has no issues. For the example in Figure 20, the sensor was subjected to controlled conditions (temperature 30 [°C] and pressure 860 [mmHg]).

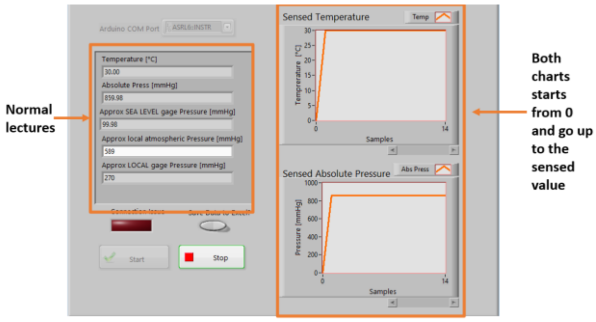


Figure 20. *Calibrated IntraSense KIT VI working well.*

4. The Calibrated IntraSense Eval Kit VI has the option to save the data by using the 'Save data?' button. All the data sensed after the button is pressed will be saved in an external CSV (Eval_Kit_Readings.csv) file. The CSV with the data will be located at the same folder than the exe. To save the data, first click on the switch and while the button is activated by the user all the data will be saved in a CSV file, to stop the data saving, the user must click the switch again.

IMPORTANT NOTE: Before opening the document, TE suggests stopping or closing the VI because while the document is open, data will not be added to the file. Every time the user starts a new test, a new document will be created with the following name structure: "Eval_Kit_Readings_x.csv", where the 'x' represents a new number starting from "1".

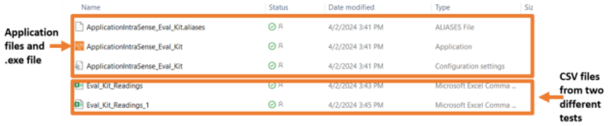
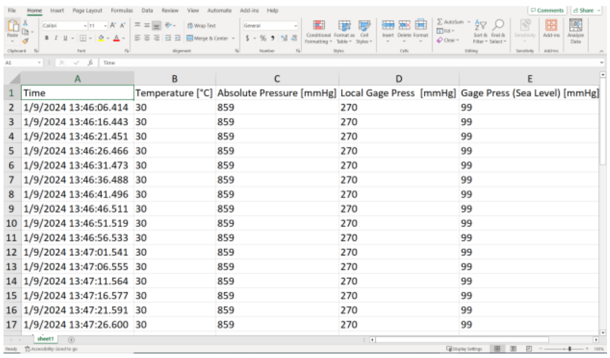


Figure 21. *Folder of the application files with two CSV files from two different tests.*

The first column of the Eval_Kit_Readings.csv document is the date and time of the test; the following four columns are the temperature and the three pressure values (Figure 22).



Time	Temperature [°C]	Absolute Pressure [mmHg]	Local Gage Press [mmHg]	Gage Press (Sea Level) [mmHg]
1/9/2024 13:46:06.414	30	859	270	99
1/9/2024 13:46:16.443	30	859	270	99
1/9/2024 13:46:21.451	30	859	270	99
1/9/2024 13:46:26.466	30	859	270	99
1/9/2024 13:46:31.473	30	859	270	99
1/9/2024 13:46:36.488	30	859	270	99
1/9/2024 13:46:41.496	30	859	270	99
1/9/2024 13:46:46.511	30	859	270	99
1/9/2024 13:46:51.519	30	859	270	99
1/9/2024 13:46:56.533	30	859	270	99
1/9/2024 13:47:01.541	30	859	270	99
1/9/2024 13:47:06.555	30	859	270	99
1/9/2024 13:47:11.564	30	859	270	99
1/9/2024 13:47:16.577	30	859	270	99
1/9/2024 13:47:21.591	30	859	270	99
1/9/2024 13:47:26.600	30	859	270	99

Figure 22. *Example of the Arduino_Eval_Kit.csv file.*

During the same test, the document will be updated every time that the user saves and stops the data. If the test is stopped a new one is started, a new document will be created.

5. To stop the application, just click on the 'Stop' button. Once the application is stopped, the last measurement of each sampled data will remain visible for the user.

6. To re-start the Calibrated IntraSense Eval Kit VI click on the white arrow at the top left of the Front-End window and press the 'Start' button again.

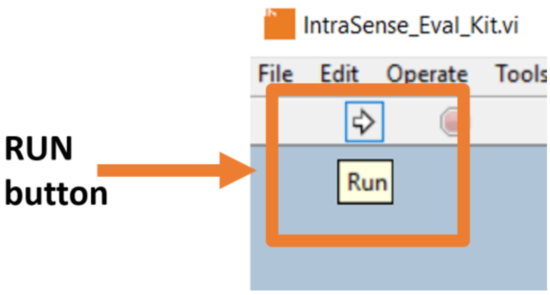


Figure 23. *Run button to re-start the application.*

If any issue is presented, please contact a TE Representative or Field Applications Engineer.

Ordering Information

Part Number	Sensor Supply Voltage (V)	Sensor Type	Trifilar Length (cm)
20031202-00	3.3	Standard	180
20031204-00		Light Shielded	
20031201-00	5	Standard	
20031203-00		Light Shielded	

your distributor
AMSYS GmbH & Co.KG
An der Fahrt 4, 55124 Mainz, Germany
Tel. +49 (0) 6131 469 875 0
info@amsys.de | www.amsys.de

te.com

Version # 01/2024