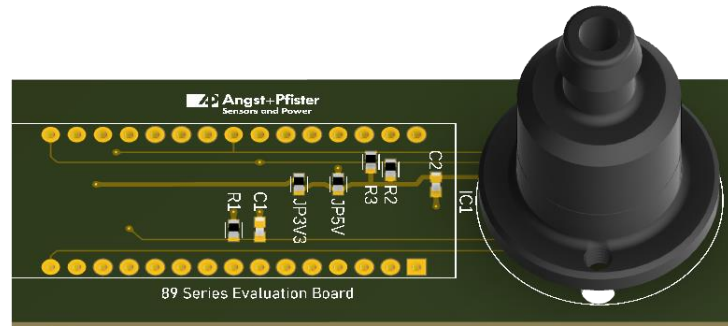


# User Manual for the 89 Series Pressure Sensor



# Contents

<b>1 Disclaimer .....</b>	<b>4</b>
<b>2 Introduction.....</b>	<b>5</b>
<b>3 Package Content.....</b>	<b>5</b>
<b>4 Setup .....</b>	<b>5</b>
4.1 Assembly .....	5
4.2 Connection .....	5
4.3 Arduino Code Editor .....	5
4.3.1 Arduino Create .....	5
4.3.2 Visual Studio Code .....	5
4.3.3. Arduino IDE .....	6
4.4 Receiving Serial Data .....	7
<b>5 Converting raw sensor values .....</b>	<b>8</b>
<b>6 Revision History .....</b>	<b>8</b>

## **1 Disclaimer**

Products mentioned in this catalogue may possibly be trademarks used only for the purposes of identification. All rights reserved.

This document may only be used by the recipient for the purpose intended. It may not be copied in whole or in part in any manner, or translated into another language, without our explicit prior consent. Subject to changes.

Copyright: Angst + Pfister Sensors and Power

Angst + Pfister Sensors and Power  
Thurgauerstrasse 66  
8050 Zürich  
Switzerland

Phone +41 44 877 35 00  
[sensorsandpower@angst-pfister.com](mailto:sensorsandpower@angst-pfister.com)  
<https://sensorsandpower.angst-pfister.com>

Michel Krapf, Zurich, Release 07.2023

## 2 Introduction

The kit presented here is designed to help implement 89 series sensors in your design. To simplify this process, this kit provides you with functioning communication with the sensor through serial (USB). One version of the code for the I<sup>2</sup>C communication is being provided to help understand possibilities to interact with sensors. The code can be altered and rewritten to test different scenarios.

Additional Information:

It is possible to use other communication protocols with the PCB, such as SPI, one wire interface, analog voltage output. However, in this manual the focus lies on sensors with I<sup>2</sup>C protocols.

## 3 Package Content

1x Arduino Nano  
1x PCB with sensor  
1x Micro – USB-A Cable  
1x Software Pack

## 4 Setup

### 4.1 Assembly

The sensor must be soldered correctly onto the PCB. The orientation is indicated by a header with the name “IC1”. Additionally, there is a wider white line outlining the place where the sensor should be soldered on.

### 4.2 Connection

Connect the PCB to the Arduino board. Then connect the Arduino to a computer.

### 4.3 Arduino Code Editor

The Arduino board can be programmed in multiple ways. For a quick start three options are listed here, while others will work just as well.

#### 4.3.1 Arduino Create

An easy way to program the board is <https://create.arduino.cc/>, since there is no full installation needed. The code can be accessed directly through the following links:

I2C\_timed\_avg:

<https://create.arduino.cc/editor/JaPew/2c243570-574d-40c9-8020-c827dd7b3be0/preview>

I2C\_timed\_request:

<https://create.arduino.cc/editor/JaPew/23f06c91-94aa-4057-9dbe-ba5980208e5d/preview>

The Website guides through how to install a plugin and then code can be run on the Arduino. However, there are limits on free compilation-time.

#### 4.3.2 Visual Studio Code

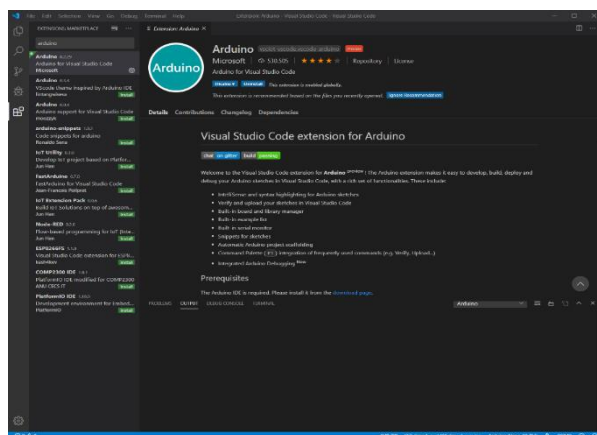


Figure 1: The "Arduino" extension must be installed for VSC to be used to upload code to an Arduino board.

Another option is “Microsoft Visual Studio Code” (not to be confused with “Microsoft Visual Studio”). It offers more of a classical IDE. Visual Studio Code (VSC) can be downloaded here:

<https://code.visualstudio.com/download>

After VSC is installed, an extension needs to be added. The extension allows VSC to communicate directly with the Arduino and updates the Arduino Libraries.



Figure 2: Settings in VSC that need to be checked and adjusted.

After installing the extensions, the blue bottom ribbon allows for some new settings. As “programmer” we recommend “AVR ISP”. When multiple Code Files are opened, “Sketch File” can be changed to whatever Sketch needs to be uploaded to the Arduino. Board Config needs to be changed to the Arduino Board used. The serial port needs to be changed to the port the Arduino is connected to.

Finally, code can be compiled and uploaded to the Arduino by pressing “Arduino: Upload” in the top right corner or pressing “Ctrl+Alt+U”.

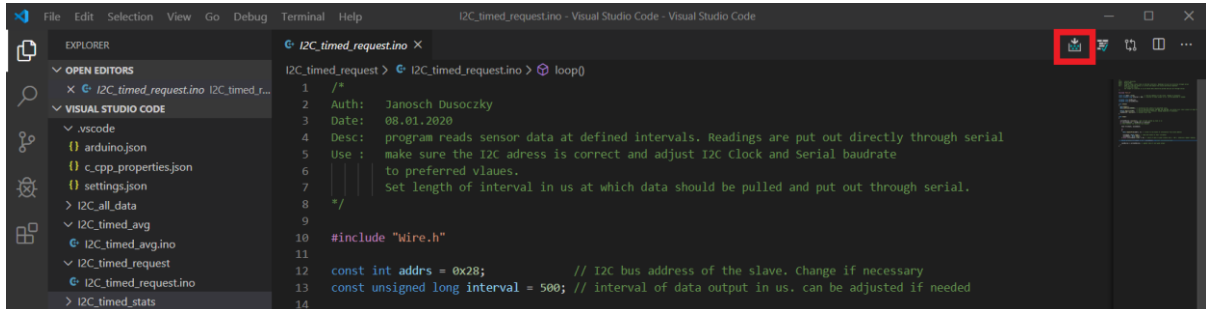
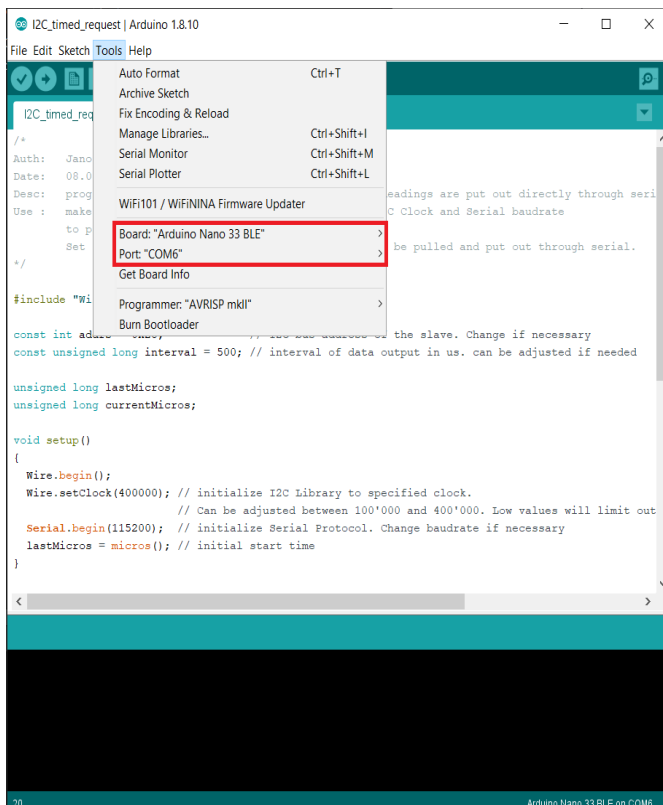


Figure 3: View of VCS with button to upload code to Arduino board highlighted.

### 4.3.3. Arduino IDE



The last option to be presented here is the Arduino IDE. It can be downloaded here: <https://www.arduino.cc/en/Main/Software>

After installation, the Arduino board has to be chosen and the serial port the Arduino is connected to needs to be defined.

Finally, code can be compiled and uploaded to the Arduino by pressing “Upload” in the top left corner.

Figure 4: Settings that need to be checked and adjusted.

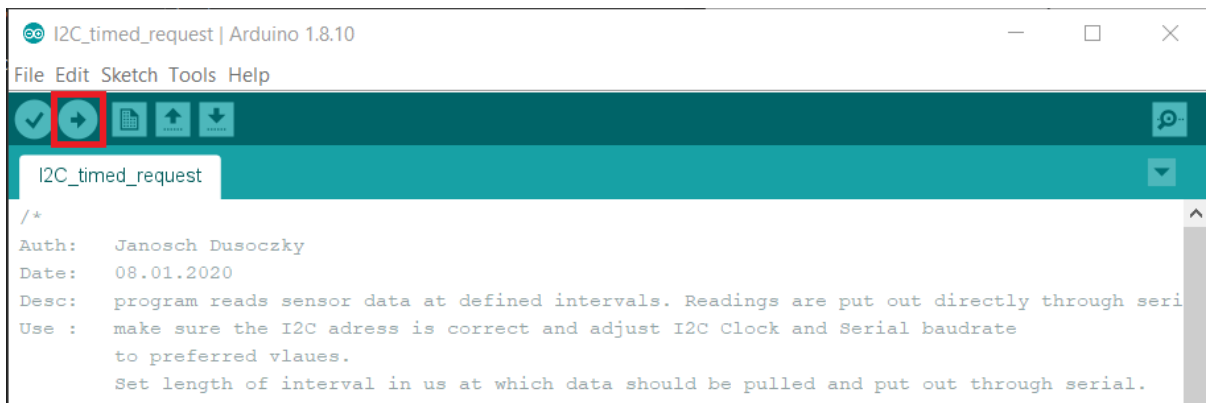


Figure 5: View of VCS with button to upload code to Arduino board highlighted.

#### 4.4 Receiving Serial Data

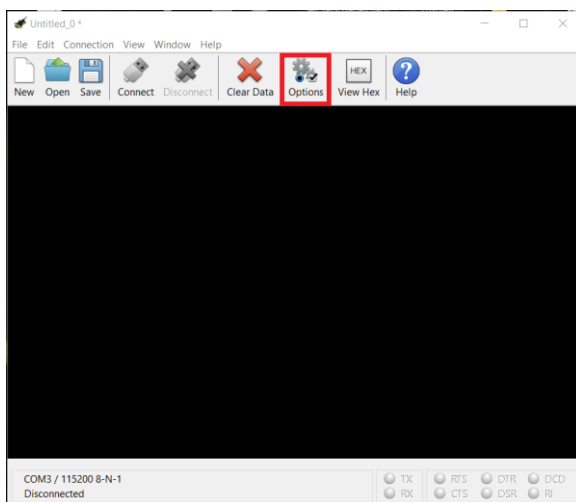


Figure 6: Under "Options", the correct port and baudrate can be set.

The Arduino can now be programmed and will send data through serial to the computer. Any program that can receive and log serial data can be used. For ease of use, we recommend "CoolTerm". CoolTerm can be downloaded here: <https://freeware.the-meiers.org/>. After installation, the correct port and baudrate have to be selected. The data received can be captured to a file by clicking "connection" -> "Capture to Textfile" -> "Start" or by pressing "Ctrl+R". An explorer window will open where location and name of the log file can be selected. When naming the file, the extension can be adjusted, for example to \*.csv for easier analysis.

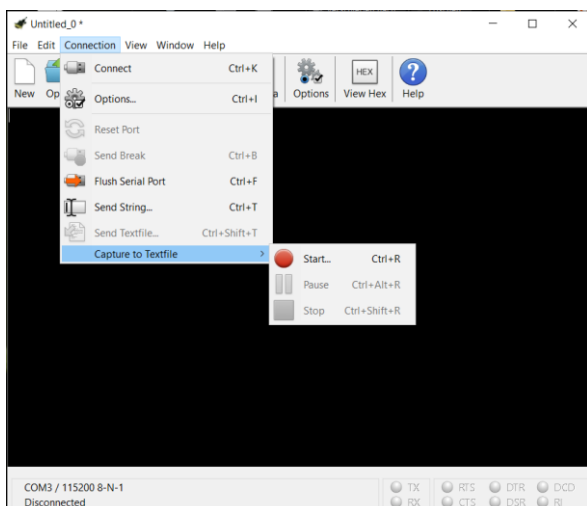


Figure 7: This way, the data that is received can be logged to a file.

## 5 Converting raw sensor values

$$\frac{\text{raw} - 0.1 * \text{adc\_fs}}{0.8 * \text{adc\_fs}} * \text{pressure\_fs} \quad \text{adc\_fs} = 2^{\text{adc\_resolution}} \quad \text{adc\_resolution} = 16\text{bit}$$

Check our [GitHub repository](#) for a concrete example of an implementation of the communication with the sensor and the conversion of raw data.

## 6 Revision History

Rev. #	Date of change	Author	Changes made
1.0	05.07.2023	Michel Krapf	Document created