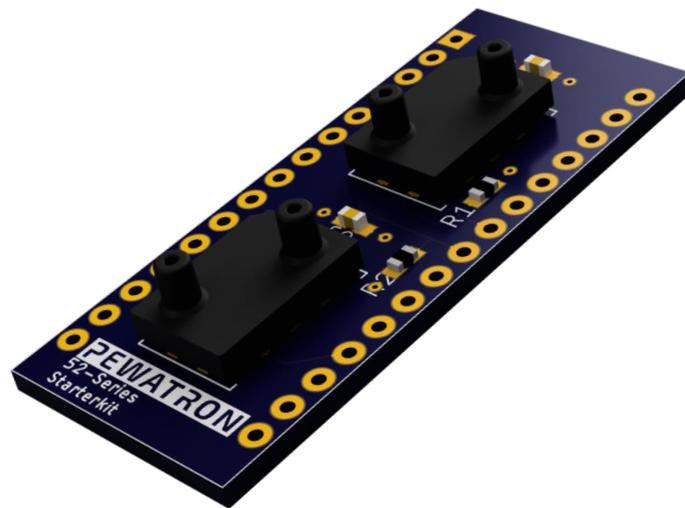


# User Manual for 52-Series Starter Kit

Rev. 1.0



# Contents

- 1 Disclaimer ..... 2
- 2 Introduction ..... 3
- 3 Package Content ..... 3
- 4 Setup ..... 3
  - 4.1 Hardware ..... 3
  - 4.2 Arduino Code Editor ..... 3
    - 4.2.1 Arduino Create ..... 3
    - 4.2.2 Visual Studio Code ..... 3
    - 4.2.3 Arduino IDE ..... 4
  - 4.3 Receiving Serial Data ..... 5
- 5 Arduino/I<sup>2</sup>C Code ..... 6
  - 5.1 I2C\_timed\_single\_raw.ino ..... 6
  - 5.2 I2C\_timed\_single\_calc.ino ..... 6
  - 5.3 I2C\_timed\_avg\_raw.ino ..... 6
  - 5.4 I2C\_timed\_avg\_calc.ino ..... 6
- 6 Revision History ..... 7

Rev. 1.0 02/2021

## 1 Disclaimer

Products mentioned in this catalogue may 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 und Pfister Sensors and Power AG

Angst und Pfister Sensors and Power AG  
Thurgauerstrasse 66  
8050 Zürich  
Switzerland

Phone +41 44 877 35 00  
sensorsandpower@angst-pfister.com  
sensorsandpower.angst-pfister.com

Janosch Dusoczky and Dario Weiss, Zurich, Release 02.2021

Rev. 1.0 02/2021

## 2 Introduction

The kit presented here is designed to help implement 52-Series I<sup>2</sup>C and analogue Sensors in your design. To simplify this process, this kit provides you with a functioning I<sup>2</sup>C communication with output through serial (USB). Different versions of code for the I<sup>2</sup>C communication are provided to help understand possibilities to interact with sensors. The code can be altered and rewritten to test different scenarios.

## 3 Package Content

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

## 4 Setup

### 4.1 Hardware

To set up the hardware, simply connect the PCB to the Arduino board. Then connect the Arduino to a computer.

### 4.2 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.

#### 4.2.1 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/dapew/3a7d543d-db58-4d40-bc6e-b06e3d925ebb/preview>

I2C\_timed\_request:

<https://create.arduino.cc/editor/dapew/e58d91fa-beb5-4fe2-b182-e0905b632cb4/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.

#### 4.2.2 Visual Studio Code

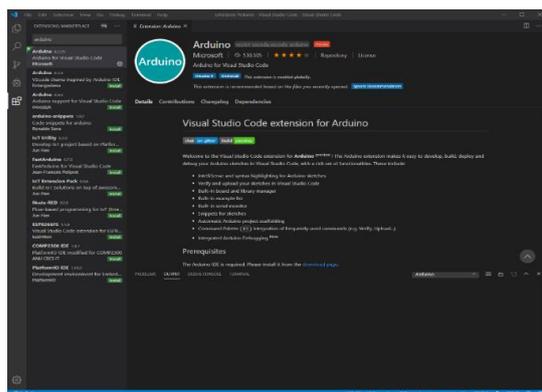


Figure 1: The "Arduino" extension must be installed for VSC to be used to upload code to an Arduino board.

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.

Rev. 1.0 02/2021



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”.

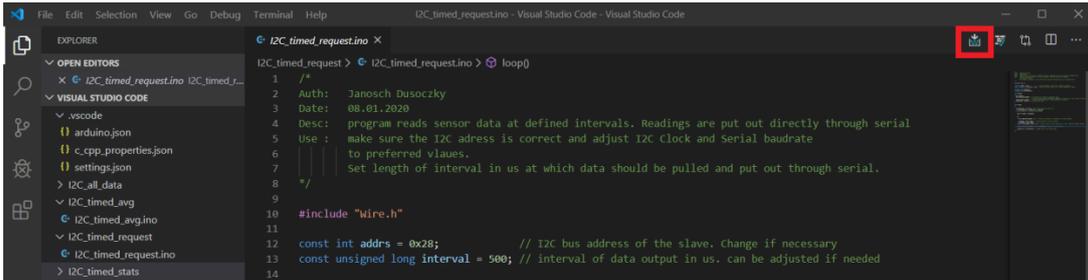


Figure 3: View of VCS with button to upload code to Arduino board highlighted.

### 4.2.3 Arduino IDE

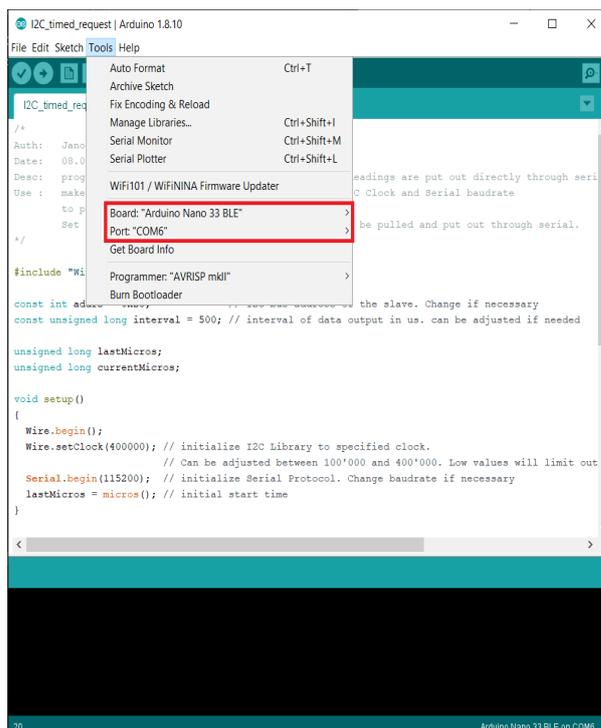


Figure 4: Settings that need to be checked and adjusted.

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 must 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.

Rev. 1.0 02/2021

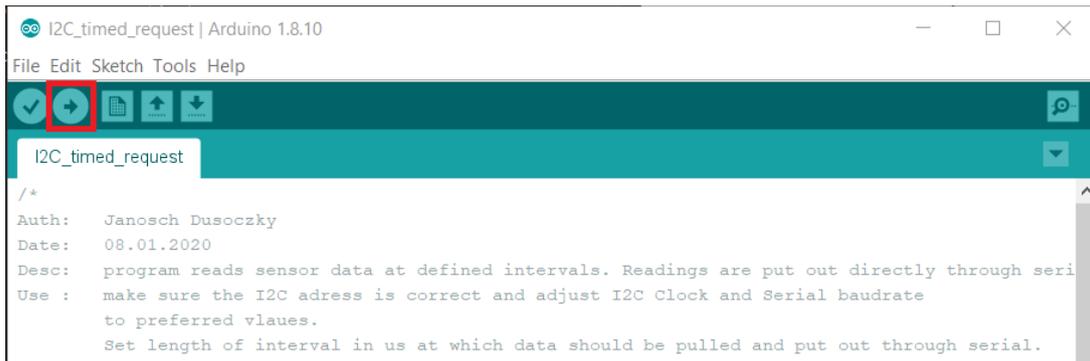


Figure 5: View of VCS with button to upload code to Arduino board highlighted.

### 4.3 Receiving Serial Data

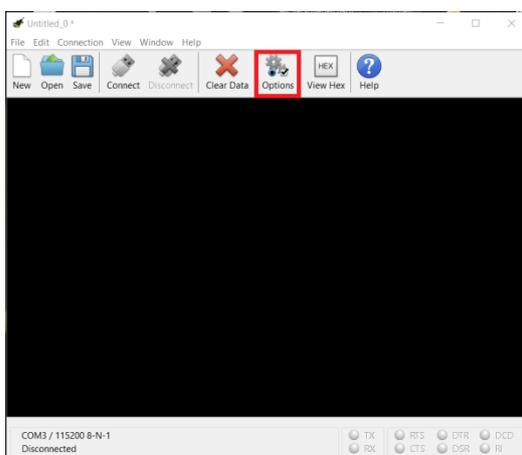


Figure 6: Under "Options", the correct port and baudrate can be set.

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 baud rate must 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 the location and a name for the log file can be selected.

When naming the file, the extension can be adjusted, for example to \*.csv for easier analysis.

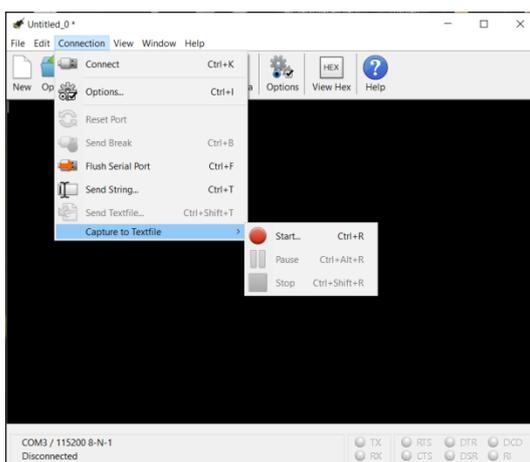


Figure 7: This way, the data that is received can be logged to a file.

Rev. 1.0 02/2021

## 5 Arduino/I<sup>2</sup>C Code

Multiple programs for different scenarios are provided. All of them can be edited and manipulated to whatever needs to be accomplished. The I<sup>2</sup>C Communication works according to our 52-Series specifications.

All programs contain a header with a brief description of what it does and what parameters can be changed without changing the program altogether.

### 5.1 I2C\_timed\_single\_raw.ino

This program fetches data from the sensor at adjustable intervals and directly outputs the raw data to serial.

### 5.2 I2C\_timed\_single\_calc.ino

This program fetches data from the sensor at adjustable intervals, calculates the pressure and outputs the data in the units specified in the header.

### 5.3 I2C\_timed\_avg\_raw.ino

This program fetches data from the sensor at approximately 2KHz. An output Interval can be defined, after which the average since the last output is calculated and sent through serial.

### 5.4 I2C\_timed\_avg\_calc.ino

This program fetches data from the sensor at approximately 2KHz. An output Interval can be defined, after which the average since the last output is calculated. The raw average is then converted into the units specified in the header and sent through serial.

Rev. 1.0 02/2021

**6 Revision History**

<b>Rev. #</b>	<b>Date of change</b>	<b>Author</b>	<b>Changes made</b>
1.0	09.02.2021	Dario Weiss	Document created

## We are here for you. Addresses and Contacts.

---

### Headquarter Switzerland:

Angst+Pfister Sensors and Power AG  
Thurgauerstrasse 66  
CH-8050 Zurich  
Phone +41 44 877 35 00  
[sensorsandpower@angst-pfister.com](mailto:sensorsandpower@angst-pfister.com)

### Office Germany:

Angst+Pfister Sensors and Power Deutschland GmbH  
Edisonstraße 16  
D-85716 Unterschleißheim  
Phone +49 89 374 288 87 00  
[sensorsandpower.de@angst-pfister.com](mailto:sensorsandpower.de@angst-pfister.com)

---

Scan here and get an overview of personal contacts!



[sensorsandpower.angst-pfister.com](https://sensorsandpower.angst-pfister.com)

---