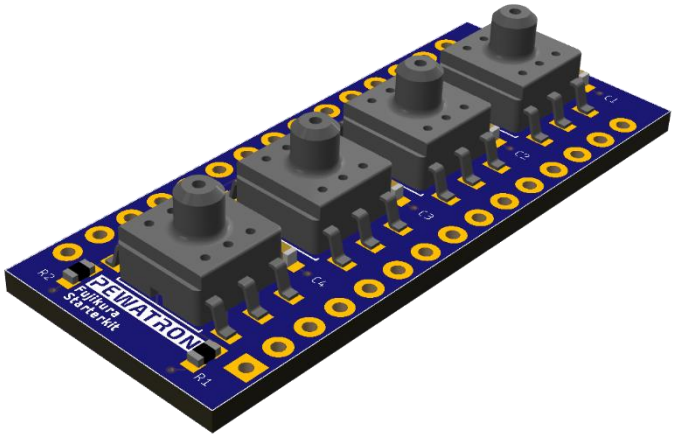
User Manual for Fujikura Starter Kit

Rev. 1.1



Contents

1 Disclaimer

Angst + Pfister Sensors and Power
Thurgauerstrasse 66
8050 Zürich
Switzerland

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

Janosch Dusoczky, Zurich, Release 06.2021

2 Introduction

The kit presented here is designed to help implement Fujikura Sensors in your design. To simplify this process, this kit provides you with a functioning sensor communication example with an output through serial (USB). The code can be altered and rewritten to test different scenarios.

3 Package Content

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

4 Setup

4.1 Assembly

First, the sensor has to be soldered onto the PCB. The spot doesn't matter. The orientation is indicated by a wider line referring to the marked edge on the sensor.

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

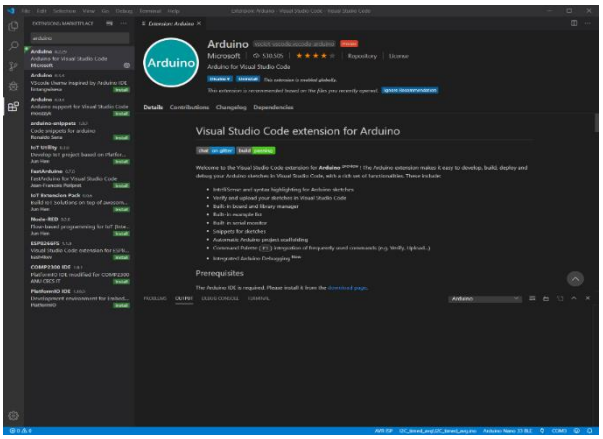


Figure 1: The "Arduino" extension must be installed for VSC to be used to upload code to an Arduino board.

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”.

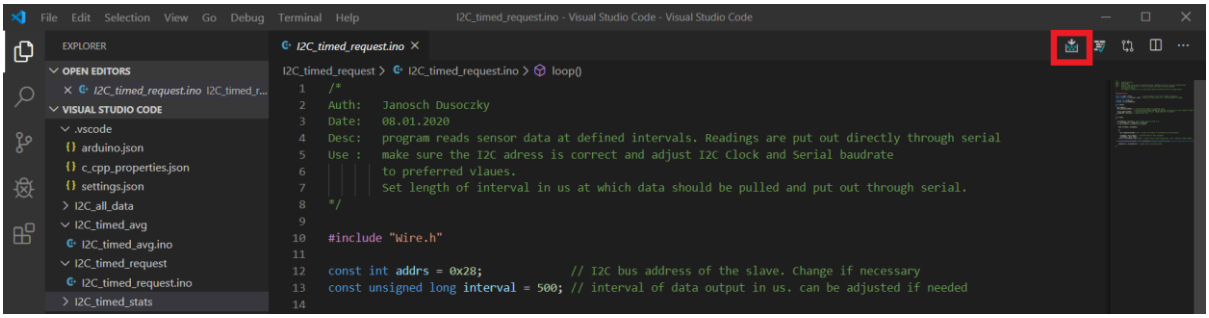


Figure 3: View of VCS with button to upload code to Arduino board highlighted.

4.3.3 Arduino IDE

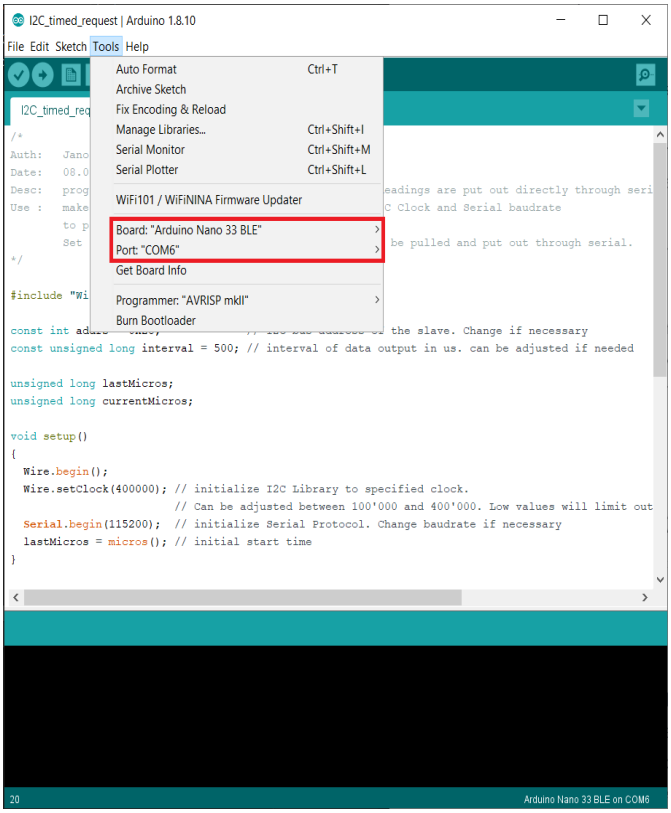


Figure 4: Settings that need to be checked and adjusted.

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 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.

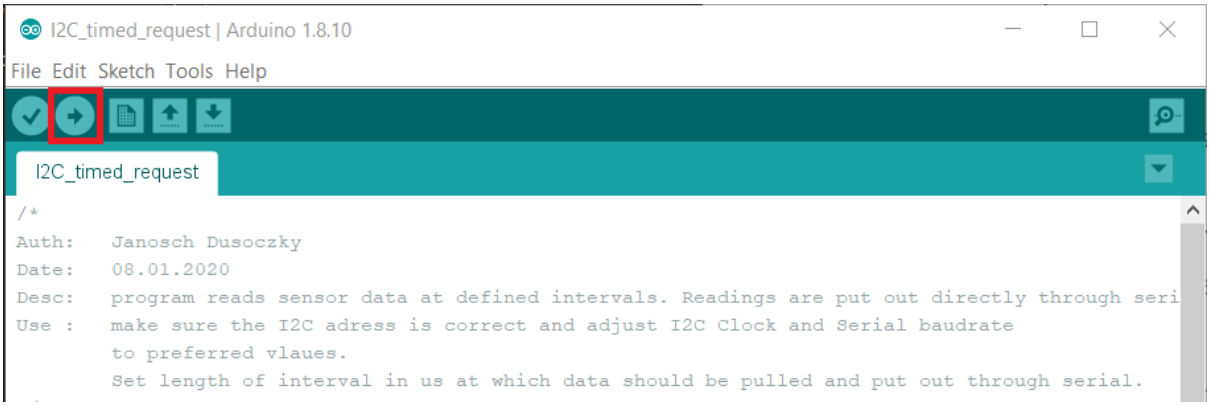


Figure 5: View of VCS with button to upload code to Arduino board highlighted.

4.4 Receiving Serial Data

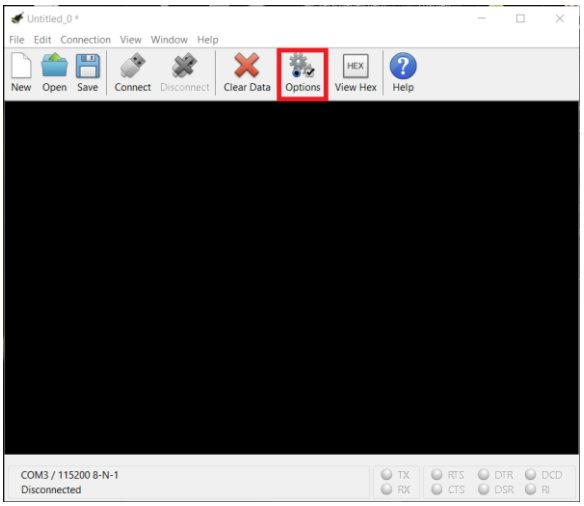


Figure 6: Under "Options", the correct port and baudrate can be set.

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 (9600) 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 a name of the log file can be selected.
When naming the file, the extension can be adjusted, for example to *.csv for easier analyzation.

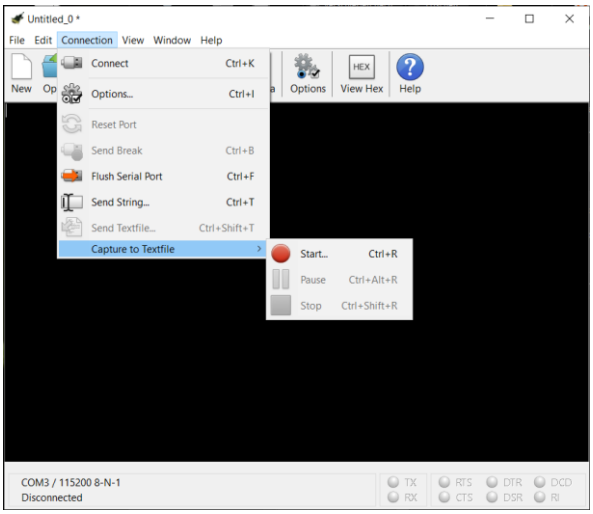


Figure 7: This way, the data that is received can be logged to a file.

5 Arduino Code

Within our repository, we have curated a folder named **ag2_starterkit_code** where you can find the initial Arduino script to help you get started with interfacing your analog AG2 pressure sensor.
Contents:

1. **AG2_Sensor_Read.ino**: This file contains a well-commented Arduino script that reads the pressure sensor data, processes it to calculate the pressure in kPa, and sends the data to the serial monitor. The script incorporates provisions to set an offset parameter, enabling the adjustment of the zero point to get more accurate readings.

Getting Started:
To begin, clone the repository or download the specific folder to your local machine. Open the **AG2_Sensor_Read.ino** file in your Arduino IDE and upload it to your Arduino board.

6 Revision History

Rev. #	Date of change	Author	Changes made
1.0	10.01.2020	Janosch Dusoczky	Document created
1.1	12.09.2023	Michel Krapf	Adjust for analog version