

Arduino Library for AMS 5812, AMS 5915, AMS 6915

Reading pressure sensor data using the I2C interface

This document describes how to use the AMS Arduino library [1] for Arduino's integrated development environment (IDE) [2]. It describes how pressure and temperature data can be read from the board mount pressure sensors AMS 5812 [3], AMS 5915 [4] and AMS 6915 [5] via the sensor's I2C interface using Arduino Uno [6] or Arduino Nano [7] and contains the reference for the AMS Arduino library and a short ready-to-use demo C source code for Arduino's IDE.

AMS 5812, AMS 5915 and AMS 6915 are board-mount pressure sensors which provide digital measurement data (pressure and temperature) through an I2C interface. These data can be read and evaluated using an Arduino Uno or Arduino Nano development board¹. To simplify reading data from AMS 5812, AMS 5915 and AMS 6915 and converting the pressure and temperature data into physical units (e.g. mbar, Pa, PSI...) Analog Microelectronics provides the AMS Arduino library for Arduino's IDE.

Although AMS 5812's, AMS 5915's and AMS 6915's data can be readout using Arduino's wire library, this method is prone to errors especially during readout and the conversion of the received data from arbitrary units into physical units. Therefore Analog Microelectronics has developed the library "AMS" for Arduino's IDE. To use this library it can be downloaded for free from www.analog-micro.com and unpacked into the Arduino/library folder. Afterwards Arduino's IDE has to be restarted. The library will be available at "Sketch -> Include Library -> Contributed libraries" then.

The library has to be included into the source code with:

```
#include <AMS.h>
```

The "AMS" library contains the following initialization sequence and functions:

Initialization sequence:

```
AMS SensorName(int sensor family, int I2C address, int minimum pressure in mbar or psi,  
int maximum pressure in mbar or psi);
```

contains the sensor family (5812, 5915 or 6915), the I2C address as well as the minimum and maximum pressure, which have to be given in the same physical units, otherwise the conversion will not work correctly. (e.g. `AMS AMSa(5812, 0x78, 0, 100);`)

Functions:

```
bool Available()
```

is used to check, if the pressure sensor responds to the given I2C address. This function returns false, if the sensor does not answer correctly, or true, if the communication with the sensor works fine.

```
void readSensor(float pressure, float temperature)
```

returns the current pressure measured by the sensor in the pressure unit given during initialization as well as the temperature measured at the sensor's pressure sensing element in °C. If an error occurred pressure and temperature will contain the value NaN.

```
float readPressure()
```

is used to read the sensor's pressure data only. The function returns the current pressure in the pressure unit given during initialization or NaN if an error occurred.

¹ For more information on how to connect AMS 5812, AMS 5915 and AMS 6915 to an Arduino Uno or Arduino Nano please see the application notes on <https://www.analog-micro.com/en/resources/applications/sensors/>.

Arduino Library for AMS 5812, AMS 5915, AMS 6915

Reading pressure sensor data using the I2C interface

```
float readTemperature()
```

is used to read the sensor's temperature data only. The returned float value contains the pressure sensor's sensing element's current temperature in °C or NaN if an error occurred.

The following code example illustrates the usage of the AMS Arduino library and the read out of pressure and temperature data from AMS 5812, AMS 5915 or AMS 6915. To use the example code only the definition of the sensor's instance has to be changed. Therein the sensor family parameter has to be adapted to the model to be read: 5812 for AMS 5812, 5915 for AMS 5915 or 6915 for AMS 6915. The I2C address has to be set to the sensor's particular I2C address and the pressure range has to be adjusted to the sensor's specific pressure range. Afterwards the code can be compiled, transmitted to Arduino Uno or Arduino Nano and the sensor's pressure and temperature data can be read using an Arduino.

```
//include the AMS library
#include <AMS.h>

//declaration of the used variables, the variables pressure and temperature have to be
float since readSensor, readPressure and readTemperature return a float value containing
the current pressure value in the given pressure unit and the current temperature in
degree celsius.
float Pressure;
float Temperature;
String DataString;
char PrintData[48];

//define the sensor's instance with the sensor's family, sensor's I2C address as well as
its specified minimum and maximum pressure. In this example an AMS 5812 pressure sensor
with a pressure range from 0 to 2068 mbar will be used. To read data from AMS 5915 or
AMS 6915 5812 has to be replaced with 5915 or 6915.
AMS AMSa(5812, 0x78,0,2068);

//initialization function which will be executed only once, when the Arduino is powered
up
void setup() {
    //set the serial port's baud rate. The COM port's standard parameters are: Data
    bits: 8, Parity: none, Stop bits: 1, Flow control: none, Baud rate: 9600
    Serial.begin(9600);
}

//main function, which will be repeated continuously until Arduino is reset or removed
from its power source and which contains the data readout from the used pressure sensor
```

Arduino Library for AMS 5812, AMS 5915, AMS 6915

Reading pressure sensor data using the I2C interface

```
void loop() {
    //Option 1: read pressure and temperature values
    //check if the pressure sensor responds to the given I2C address
    if (AMSA.Available() == true) {
        //read the sensor's temperature and pressure data. Please note that the
        //temperature is measured inside the sensor's package and contains the sen-
        //sor's self-heating as well as the ambient temperature
        AMSA.readSensor(Pressure, Temperature);

        //check if an error occurred leading to the function returning a NaN
        if (isnan(Pressure) || isnan(Temperature)) {
            //write an error message on the serial port
            Serial.write("Please check the sensor family name.");
        } else {
            //put the data into a string
            DataString = String(Pressure) + " mbar " + String(Temperature) + " "
            + (char)176 + "C \n";
            //convert string into CharArray
            DataString.toCharArray(PrintData, 48);
            //write the sensor's data on the serial port
            Serial.write(PrintData);}
    } else {
        //tell the user that there is something wrong with the communication be-
        //tween Arduino and the sensor.
        Serial.write("The sensor did not answer.");}
    //wait for 1 s until reading new data from the Sensor
    delay(1000);

    //Option 2: read pressure values only
    if (AMSA.Available() == true) {
        //read the sensor's pressure data only
        Pressure = AMSA.readPressure();
        if (isnan(Pressure)) {
            Serial.write("Please check the sensor family name.");
        } else {
            DataString = String(Pressure) + " mbar \n";
            DataString.toCharArray(PrintData, 48);
```

Arduino Library for AMS 5812, AMS 5915, AMS 6915

Reading pressure sensor data using the I2C interface

```
        Serial.write(PrintData);}

    } else {
        Serial.write("The sensor did not answer.");}
    delay(1000);

    //Option 3: read temperature values only
    if (AMSA.Available() == true) {
        //read the sensor's temperature data in degree celsius
        Temperature = AMSA.readTemperature();

        if (isnan(Temperature)) {
            Serial.write("Please check the sensor family name."); //
        } else {
            DataString = String(Temperature) + " " + (char)176 + "C \n";
            DataString.toCharArray(PrintData, 48);
            Serial.write(PrintData);}
        } else {
            Serial.write("The sensor did not answer.");}
        delay(1000);
    }
}
```

References:

- 1.) AMS Arduino library "AMS.zip" (see https://www.analog-micro.com/_pages/sens/ams5812/AMS.zip)
- 2.) Aduino's IDE (<https://www.arduino.cc/en/Main/Software>)
- 3.) AMS 5812's data sheet
(see https://www.analog-micro.com/_pages/sens/ams5812/ams5812_data_sheet.pdf)
- 4.) AMS 5915's data sheet
(see https://www.analog-micro.com/_pages/sens/ams5915/ams5915_data_sheet.pdf)
- 5.) AMS 6915's data sheet
(see https://www.analog-micro.com/_pages/sens/ams6915/ams6915_data_sheet.pdf)
- 6.) Arduino Uno (<https://store.arduino.cc/arduino-uno-rev3>)
- 7.) Arduino Nano (<https://store.arduino.cc/usa/arduino-nano>)