

```

#include <Wire.h>
#include <AMS5915.h>

AMS5915 ams;
const float AMS5915_0050_D_P_MIN = 0.0; // minimum pressure, millibar
const float AMS5915_0050_D_P_MAX = 10.0; // maximum pressure, millibar

void setup(void)
{
  Serial.begin(9600);

  // The address can be changed making the option of connecting multiple devices
  ams.getAddr_AMS5915(AMS5915_DEFAULT_ADDRESS);

  ams.begin();
  delay(500);
}

void loop(void)
{
  byte error;
  int8_t address;

  address = ams.ams_i2cAddress;
  // The i2c_scanner uses the return value of
  // the Write.endTransmission to see if
  // a device did acknowledge to the address.
  Wire.beginTransmission(address);
  error = Wire.endTransmission();
  if (error == 0)
  {
    ams.Measure_PressureAndTemperature(AMS5915_0050_D_P_MIN, AMS5915_0050_D_P_MAX);
    float cTemp, fTemp, pressure, kPa, hg, PSI;

    Serial.println("Getting Readings from AMS5915_0050-D");
    Serial.println(" ");
    // Read and print out the Pressure, convert it to mbar, PSI, kPa, mmHg
    pressure = ams.getPressure();
    PSI = pressure * 0.0145038;
    kPa = pressure / 10;
    hg = pressure * 76.0 / 101.325;

    // Read and print out the temperature, then convert to C and F scales
    cTemp = ams.getTemperature();
    fTemp = cTemp * 1.8 + 32;

    // Output data to screen
    Serial.print("Digital Pressure Reading: ");
    Serial.print(pressure);
    Serial.println(" mbar");
    Serial.print("Digital Pressure Reading: ");
    Serial.print(PSI);
    Serial.println(" PSI");
    Serial.print("Digital Pressure Reading: ");
    Serial.print(kPa);
    Serial.println(" kPa");
    Serial.print("Digital Pressure Reading: ");
    Serial.print(hg);
    Serial.println(" mmHg");
    Serial.print("Temperature Reading in Celsius: ");
    Serial.print(cTemp);
    Serial.println(" °C");
    Serial.print("Temperature Reading in Fahrenheit: ");
    Serial.print(fTemp);
    Serial.println(" °F");
    Serial.println(" ");
    Serial.println(" ***** ");
    Serial.println(" ");
  }
  else
  {
    Serial.println("AMS5915_0050-D Disconnected!");
    Serial.println(" ");
    Serial.println(" ***** ");
    Serial.println(" ");
  }

  delay(1000);
}

```

Libraries section (can add additional if you want the program to do additional functions beyond the sample program)

Allows the program to understand basic unique commands specially created for this sensor

Baud rate

i2c address

Sends signals (in this case 10 times per second) to the sensor so the sensor can report back 10 data samples (best not to change the delay value)

Don't edit any of this or it won't execute the program

- Instructions to add Kelvin (or any other variable):
1. Add K to the right of PSI in the float line
 2. Add the conversion below the fTemp conversion line
 $K = fTemp + 273.15;$
 3. Add appropriate print statement below with the rest of the print statements

Commands transmission from the sensor

floats define numerical values (these you can edit and add more)

Initial print statement (is displayed at the very top of the data string and can say whatever you would like)

Define what your floats equal (assigning values to them and doing unit conversions)

Like pressure, this command pulls temperature data from the sensor and the following line converts that data to °F (to get Kelvin, see above)

Print statements (what the monitor displays after executing the program)
To neglect the Celsius data, change the print statements to:
//Serial.print("Temperature Reading in Celsius: ");
//Serial.print(cTemp);
//Serial.println(" °C");

To add a print statement to print "Hello World" in the data string, type:
Serial.print("Hello World");

Each of these print statements have 3 sections:
Serial.print("Digital Pressure Reading: "); Titles the data
Serial.print(pressure); Displays the data value from the sensor
Serial.println(" mbar "); Assigns units to the data value

Some helpful notes:
- Spaces in the print statements carry over into the serial data
-To add a blank row in the data string, type: Serial.println(" ");

Else print statement (alternative statement is printed if connection to the sensor is lost)

sample rate (1000 = 1 sample per second, 500 = 10 samples per second)