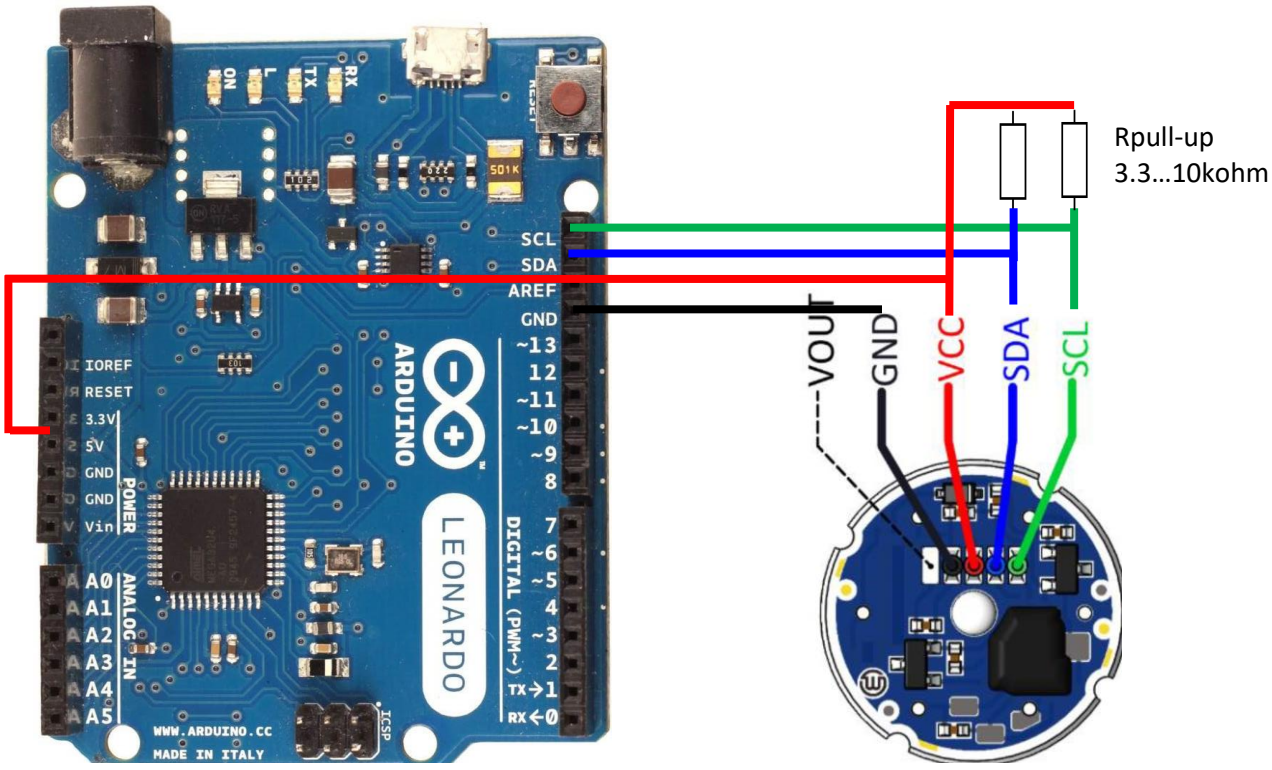
Setup information:

- Power supply voltage must be in the same range used during calibration.
  - If calibrated at 5.0V sensor must be used between 3.6V to 5.5V
  - If calibrated at 3.3V sensor must be used between 2.7V to 3.6V
- I2C address is 0x78
- Secondary I2C address is normally disabled. Can be modify during or after calibration.
- 10 kohm pull-up resistors are needed between:
  - Vcc and SDA
  - Vcc and SCL
- Pressure information is based on 15bit ADC
  - 5% of 15bit is the lowest value associated to the lowest pressure (typically 0bar)
  - 95% of 15bit is the highest value associated to the highest pressure (Full scale)
- Electrical connection (standard product)
  - Red = Vcc
  - Black = GND
  - Blue = SDA
  - Green = SCL



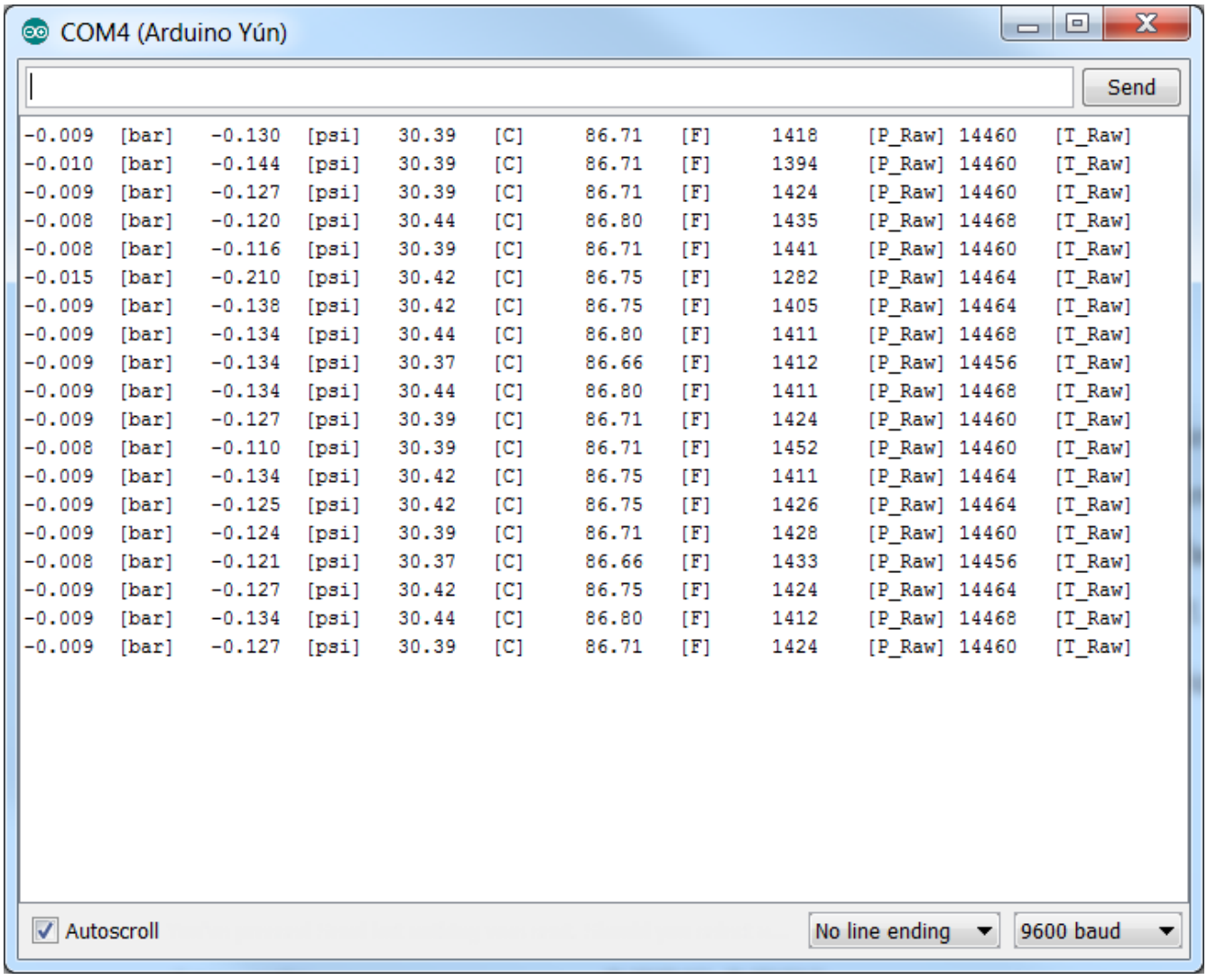
Example Arduino code:

```
/*****************************************************/
/***  ME782 read out pressure and temperature    ***/
/***         M.Croci - Metallux SA - 03.04.2015          ***/
/***    Compile, upload and open serial monitor    ***/
/*****************************************************/

/*** Library ***/
#include <Wire.h>                       //Library for I2C

/*** Variables ***/
uint16_t raw_pressure;                  //Pressure raw data
uint16_t raw_temperature;               //Temperature raw data
float bar, psi;                         //Pressure in bar, Pressure in PSI,
float tempCelsius, tempFahrenheit;      //Temperature in °C, Temperature in °F
byte P_MSB, P_LSB, T_MSB, T_LSB;        //Very important to use byte for this variables.

/*** Constant ***/
byte zaddr = 0x78;                      //I2C communication address
const float Pmin = 0;                   //Write here the minimum pressure value (defined by sensor calibration)
const float Pmax = 1.2;                 //Write here the full scale pressure value (defined by sensor calibration)
const float P_A = 0.05;                 //this is constant (defined by sensor calibration)
const float P_B = 0.95;                 //this is constant (defined by sensor calibration)

const float Tmin = -40.0;               //this is constant (defined by sensor calibration)
const float Tmax = 125.0;               //this is constant (defined by sensor calibration)
const float T_A = 0.1;                  //this is constant (defined by sensor calibration)
const float T_B = 0.9;                  //this is constant (defined by sensor calibration)

/*** Setup ***/
void setup()                            // SETUP is executed only one time at the beginning
{
 Wire.begin();                          //join i2c bus (address optional for master)
 Serial.begin(9600);                    //start serial for output
 delay(2000);                           //wait port setup
}
```

```
/*** Main ***/
void loop()                                          // LOOP is executed continuously
{
 Wire.requestFrom(zaddr, byte(4));                   // request 4 bytes from slave device #2
   while(Wire.available())                           // slave may send less than requested
   {
      P_MSB = Wire.read();                           // receive the Pressure MSB
      P_LSB = Wire.read();                           // receive the Pressure LSB
      T_MSB = Wire.read();                           // receive the Temperature MSB
      T_LSB = Wire.read();                           // receive the Temperature LSB

      raw_pressure = P_MSB<<8;                       //Shift MSB and Concatenate LSB (pressure)
      raw_pressure += P_LSB;

      raw_temperature = T_MSB<<8;                    //Shift MSB and Concatenate LSB (temperature)
      raw_temperature += T_LSB;

      bar = raw2bar(raw_pressure);                   //convert pressure raw value to bar
      psi = bar*14.5037738;                          //Conversion to psi

      tempCelsius = raw2celsius(raw_temperature);    //convert temperature raw value to celsius
      tempFahrenheit = tempCelsius*1.8+32;           //conversion in °F

      //Print the values of pressure, temperature and raw data to the "serial monitor"
      Serial.print(bar,3);                  //Pressure (bar)
      Serial.print("\t");                   //Tab (I use tab because you can copy the data in excel directly)
      Serial.print("[bar]");                //Measure unit
      Serial.print("\t");                   //Tab

      Serial.print(psi,3);
      Serial.print("\t");
      Serial.print("[psi]");
      Serial.print("\t");

      Serial.print(tempCelsius,2);
      Serial.print("\t");
      Serial.print("[C]");
      Serial.print("\t");

      Serial.print(tempFahrenheit,2);
      Serial.print("\t");
      Serial.print("[F]");
      Serial.print("\t");

      Serial.print(raw_pressure, DEC);
      Serial.print("\t");
      Serial.print("[P_Raw]");
      Serial.print("\t");

      Serial.print(raw_temperature, DEC);
      Serial.print("\t");
      Serial.println("[T_Raw]");

   }
 delay(500);  //Delay between measures
```

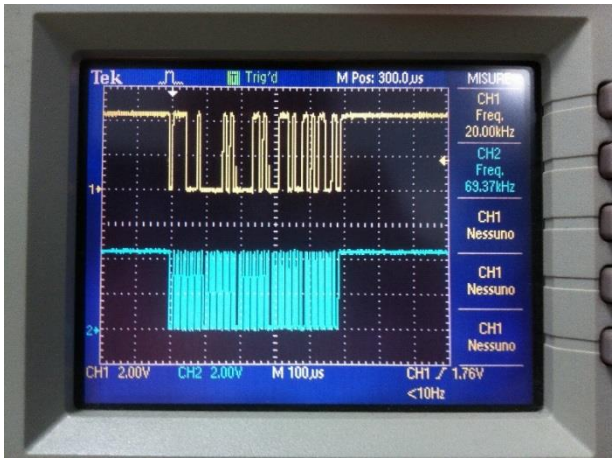
```
 }

/*** Function definition ***/
// conversions data (raw pressure to bar)
float raw2bar(uint16_t raw)
{
        return  (raw/32767.0)*((Pmax-Pmin)/(P_B-P_A))+(Pmin-((Pmax-Pmin)/(P_B-P_A))*P_A);
}

// conversions data (raw temperature to celsius)
float raw2celsius(uint16_t raw)
{
        return  (raw /32767.0)*((Tmax-Tmin) / (T_B-T_A))+(Tmin - ((Tmax-Tmin) / (T_B-T_A))*T_A);
}
```

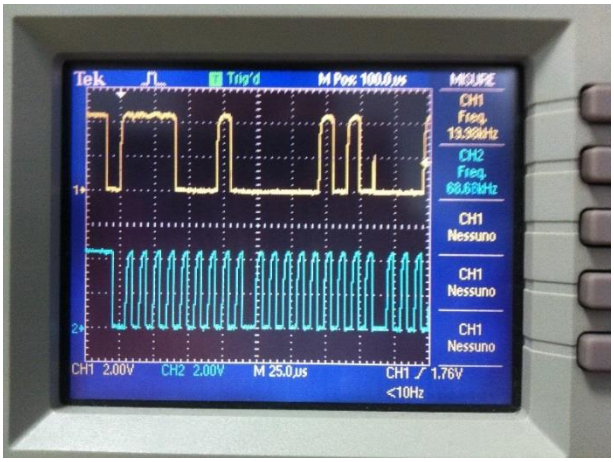
Output example



Timing example

Yellow = SDA
Blue = SCL





Nr.	Parameter	Symbol	min	typ	Max	Unit	Conditions
1	SCL clock frequency *	f_{SCL}			400	kHz	$f_{OSC} \geq 2MHz$
2	Bus free time between start and stop condition	t_{I2C_BF}	1.3			µs	
3	Hold time start condition	$t_{I2C\_HD\_STA}$	0.6			µs	
4	Setup time repeated start condition	$t_{I2C\_SU\_STA}$	0.6			µs	
5	Low period SCL/SDA	t_{I2C_L}	1.3			µs	
6	High period SCL/SDA	t_{I2C_H}	0.6			µs	
7	Data hold time	$t_{I2C\_HD\_DAT}$	0			µs	
8	Data setup time	$t_{I2C\_SU\_DAT}$	0.1			µs	
9	Rise time SCL/SDA	t_{I2C_R}			0.3	µs	
10	Fall time SCL/SDA	t_{I2C_F}			0.3	µs	
11	Setup time stop condition	$t_{I2C\_SU\_STO}$	0.6			µs	
12	Noise interception SDA/SCL	t_{I2C_NI}			50	ns	Spike suppression