

## MEAS MS8607 XPLAINED PRO BOARD

Digital Pressure and Humidity  
Digital Component Sensor (DCS) Development Tools

### Performance

- ◆ 0% to 100% relative humidity range
- ◆ -40°C to 125°C temperature range
- ◆ Very low power consumption
- ◆ Operates from 1.5V to 3.6V
- ◆ Fast response time – 5 seconds typical
- ◆ Built-in heater for fast recovery from saturation
- ◆ Recovers fully from condensation
- ◆ Fast conversion time – 14ms typical

### Features

- ◆ 20-pin Xplained Pro compatible connector
- ◆ I<sup>2</sup>C interface
- ◆ Xplained Pro hardware identification chip
- ◆ Atmel Studio 6 Project available for download
- ◆ µC C code available for download
- ◆ Selectable 16-24 bits resolution for pressure
- ◆ Selectable 8-12 bit resolution for humidity
- ◆ Selectable 12-14 bit resolution for temperature

The MS8607 Xplained Pro provides the necessary hardware to interface the MS8607 digital pressure, relative humidity and temperature sensor to any system that utilizes Xplained Pro compatible expansion ports configurable for I<sup>2</sup>C communication. The MS8607 sensor is a self-contained pressure, humidity and temperature sensor that is fully calibrated during manufacture. The sensor can operate from 1.5V to 3.6V. The MS8607 is ideal for weather station applications embedded into compact devices and any applications in which pressure, humidity and temperature monitoring is required.

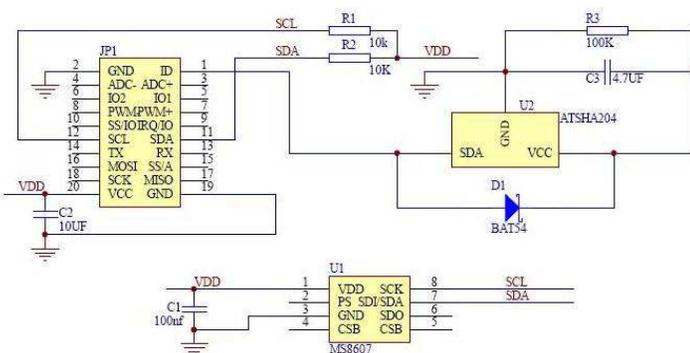
Refer to the MS8607 data sheet for detailed information regarding operation of the IC:

<http://www.te.com/usa-en/product-CAT-BLPS0018.html>

### Specifications

- ◆ Operating pressure range: 300 to 1200 mbar
- ◆ Measures relative humidity from 0% to 100%
- ◆ Measures temperature from -40°C to 125°C
- ◆ Extended pressure range 10 to 2000 mbar
- ◆ Fast response time
- ◆ I<sup>2</sup>C communication
- ◆ Very low power consumption

### Schematic

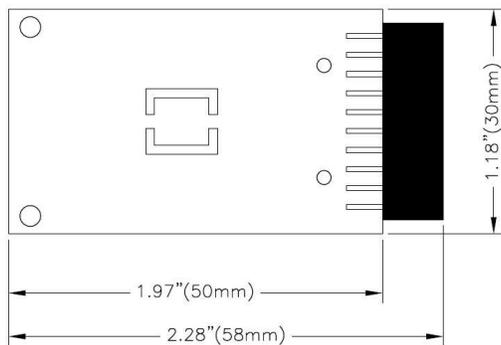


## Connector Pin Assignments (I<sup>2</sup>C Communications)

### System Plug

Connector JP1					
Pin No.	Signal	Description	Pin No.	Signal	Description
1	ID	Hardware identification	11	SDA	TWI Serial Data
2	GND	Ground	12	SCL	TWI Serial Clock
3	N/C	Not Connected	13	N/C	Not Connected
4	N/C	Not Connected	14	N/C	Not Connected
5	N/C	Not Connected	15	N/C	Not Connected
6	N/C	Not Connected	16	N/C	Not Connected
7	N/C	Not Connected	17	N/C	Not Connected
8	N/C	Not Connected	18	N/C	Not Connected
9	N/C	Not Connected	19	GND	Ground
10	N/C	Not Connected	20	Vdd	Power Supply

## Dimensions (mm)



## Detailed Description

### I<sup>2</sup>C Interface

The peripheral module can interface to the host being plugged directly into an Xplained Pro extension port (configured for I<sup>2</sup>C) through connector JP1.

### External Control Signals

The IC operates as an I<sup>2</sup>C slave using the standard 2 wire I<sup>2</sup>C connection scheme. The IC is controlled either by the host (through the Xplained Pro connector). In cases where one or more of the SCL and SDA signals are driven from an external source, 10k resistors R1, R2 provide pull-up. However, this also increases the apparent load to the external driving source. If the external source is not capable of driving these loads (10k), they should be removed.

### Reference Materials

The complete software kit is available for download at: [Link to MS8607\\_CPROJ.zip](#)

## Drivers & Software

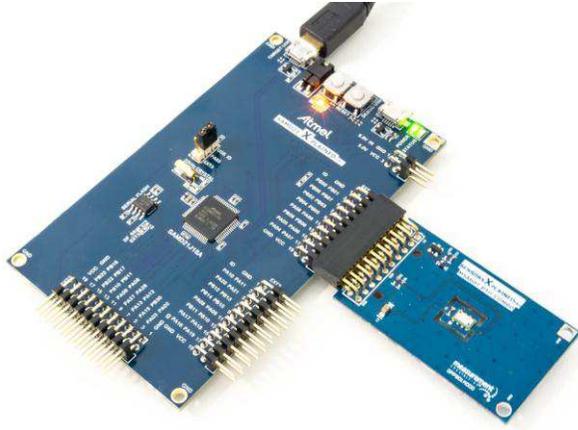
Detailed example software and drivers are available that execute directly without modification on a number of development boards that support an integrated or synthesized microprocessor. The download contains several source files intended to accelerate customer evaluation and design. The source code is written in standard ANSI C format, and all development documentation including theory/operation, register description, and function prototypes are documented in the interface file.

## Functions Summary

Enumerations	
enum	<b>ms8607_humidity_i2c_master_mode</b> { ms8607_i2c_hold, ms8607_i2c_no_hold }
enum	<b>ms8607_status</b> { ms8607_status_ok, ms8607_status_no_i2c_acknowledge, ms8607_status_i2c_transfer_error, ms8607_status_crc_error, ms8607_status_heater_on_error }
enum	<b>ms8607_humidity_resolution</b> { ms8607_humidity_resolution_12b = 0, ms8607_humidity_resolution_8b, ms8607_humidity_resolution_10b, ms8607_humidity_resolution_11b }
enum	<b>ms8607_battery_status</b> { ms8607_battery_ok, ms8607_battery_low }
enum	<b>ms8607_heater_status</b> { ms8607_heater_off, ms8607_heater_on }
enum	<b>ms8607_pressure_resolution</b> { ms8607_pressure_resolution_osr_256 = 0, ms8607_pressure_resolution_osr_512, ms8607_pressure_resolution_osr_1024, ms8607_pressure_resolution_osr_2048, ms8607_pressure_resolution_osr_4096, ms8607_pressure_resolution_osr_8192 }
Functions	
void	<b>ms8607_init</b> (void) Configures the SERCOM I2C master to be used with the MS8607 device.
bool	<b>ms8607_is_connected</b> (void) Reset the MS8607 device.
enum ms8607_status	<b>ms8607_reset</b> (void) Reset the MS8607 device.
enum ms8607_status	<b>ms8607_set_humidity_resolution</b> (enum ms8607_humidity_resolution) Set Humidity sensor ADC resolution.
void	<b>ms8607_set_pressure_resolution</b> (enum ms8607_pressure_resolution) Set Pressure sensor ADC resolution.
void	<b>ms8607_set_humidity_i2c_master_mode</b> (enum ms8607_humidity_i2c_master_mode) Set I2C master mode.
enum ms8607_status	<b>ms8607_read_temperature_pressure_humidity</b> (float *, float *, float *) Reads the temperature, pressure and relative humidity value.
enum ms8607_status	<b>ms8607_get_battery_status</b> (enum ms8607_battery_status *) Provide battery status.
enum ms8607_status	<b>ms8607_enable_heater</b> (void) Enable heater.
enum ms8607_status	<b>ms8607_disable_heater</b> (void) Disable heater.
enum ms8607_status	<b>ms8607_get_heater_status</b> (enum ms8607_heater_status *) Get heater status.
enum ms8607_status	<b>ms8607_get_compensated_humidity</b> (float, float, float *) Returns result of compensated humidity Note : This function shall only be used when the heater is OFF. It will return an error otherwise.
enum ms8607_status	<b>ms8607_get_dew_point</b> (float, float, float *) Returns the computed dew point Note : This function shall only be used when the heater is OFF. It will return an error otherwise.

## Project Setup

This project is based on ATSAM20J18 board with Measurement Specialties Xplained Pro extension board connected to EXT1 pad as shown on figure below.



## Running the Application

1. Download the MS8607 Xplained Pro example package on [TE Website](#)
2. Decompress the archive file
3. Open the .cproj project file with Atmel Studio 6
4. You will now be able to build the MS8607 example project - 
5. Finally, run the build result on your Xplained Pro Board - 

## Application Code

This section is intended to provide a basic example of functionality.

```
/**
 * \file main.c
 *
 * \brief MS8607 Pressure monitoring application file
 *
 * Copyright (c) 2014 Measurement Specialties. All rights reserved.
 *
 */

#include <asf.h>

float temperature;
float pressure;
float relative_humidity;
float compensated_humidity;
float dew_point;

int main (void)
{
```

```
enum ms8607_status status;
float last_temperature = 0;
float variation = 0;
uint8_t n=0;
enum ms8607_heater_status heater;

system_init();
delay_init();

// Configure device and enable
ms8607_init();

if( !ms8607_is_connected() )
    return -1;

// Reset MS8607
status = ms8607_reset();
if( status != ms8607_status_ok)
    return -1;

// Configure resolution
status = ms8607_set_humidity_resolution(ms8607_humidity_resolution_12b);
if( status != ms8607_status_ok)
    return -1;

ms8607_set_pressure_resolution(ms8607_pressure_resolution_osr_8192);

// Monitor temperature every 500ms
while (1) {

    if( n&1 )
        ms8607_set_humidity_i2c_master_mode(ms8607_i2c_hold);
    else
        ms8607_set_humidity_i2c_master_mode(ms8607_i2c_no_hold);

    // Enable heater for 10s
    if( (n==10) ) {
        status = ms8607_enable_heater();
        if( status != ms8607_status_ok)
            return -1;
    }

    // Disable heater after 20s
    if( (n==20) ) {
        status = ms8607_disable_heater();
        if( status != ms8607_status_ok)
            return -1;
    }

    // Check heater status
```

```
    status = ms8607_get_heater_status(&heater);
    if( status != ms8607_status_ok)
        return -1;

    status = ms8607_read_temperature_pressure_humidity(&temperature, &pressure, &relative_humidity);
    if( status != ms8607_status_ok)
        return -1;

    status = ms8607_get_compensated_humidity(temperature,relative_humidity,&compensated_humidity);
    status = ms8607_get_dew_point(temperature,relative_humidity,&dew_point);

    variation += temperature - last_temperature;

    // Look for significant temperature variation
    if ( variation >= 0.5 ) {
        // Yes, so turn LED on.
        port_pin_set_output_level(LED_0_PIN, LED_0_ACTIVE);
        variation = 0;
    } else if ( variation <= -0.5 ) {
        // No, so turn LED off.
        port_pin_set_output_level(LED_0_PIN, LED_0_INACTIVE);
        variation = 0;
    }

    delay_ms(1000);
    last_temperature = temperature;
    n++;
    if(n==50) n=0;
}
}
```

## MEAS MS8607 XPLAINED PRO BOARD

Digital Pressure and Humidity DCS Development Tools

### Ordering Information

Description	Part Number
MEAS MS8607 XPLAINED PRO BOARD	DPP901A000

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

#### [te.com/sensorsolutions](http://te.com/sensorsolutions)

Measurement Specialties, Inc., a TE Connectivity company.

MEAS, Measurement Specialties (logo), TE Connectivity and TE connectivity (logo) are trademarks. All other logos, products and/or company names referred to herein might be trademarks of their respective owners.

The information given herein, including drawings, illustrations and schematics which are intended for illustration purposes only, is believed to be reliable. However, TE Connectivity makes no warranties as to its accuracy or completeness and disclaims any liability in connection with its use. TE Connectivity's obligations shall only be as set forth in TE Connectivity's Standard Terms and Conditions of Sale for this product and in no case will TE Connectivity be liable for any incidental, indirect or consequential damages arising out of the sale, resale, use or misuse of the product. Users of TE Connectivity products should make their own evaluation to determine the suitability of each such product for the specific application.

© 2016 TE Connectivity Ltd. family of companies All Rights Reserved.

#### PRODUCT SHEET

MEAS France SAS,  
a TE Connectivity company.  
Impasse Jeanne Benozzi CS 83 163  
31027 Toulouse Cedex 3, FRANCE  
Tel: +33 (0) 5 820 822 02  
Fax: +33 (0) 5 820 821 51  
[customercare.tlse@te.com](mailto:customercare.tlse@te.com)