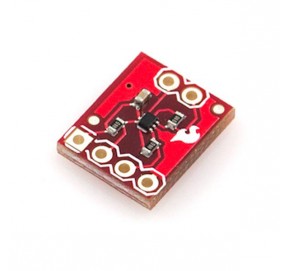
**Module Sparkfun** SEN-09418 **(TMP102)**

**Remarques pour les PPE**

Ce module à très faible consommation basé sur un TMP102 mesure la température avec une précision de 0,5°C et communique l'information via une interface I2C.

****Sparkfun TMP102 (I2C)

**Caractéristiques**

Alimentation: 1,4 à 3,6 Vcc

Consommation: 10 µA maxi (1 µA en veille)

Plage de mesure: -40°C à +125°C

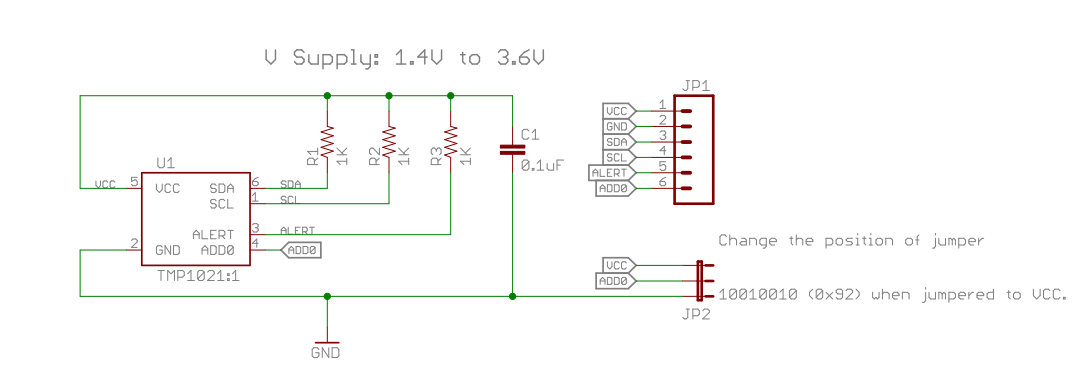
Précision: 0,5 °C (de -25°C à +85°C)

Résolution: 0,0625°C

Interface série 2 fils

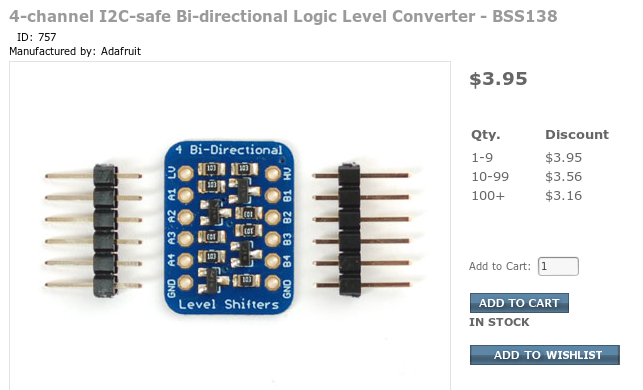
Dimensions: 13 x 10 mm

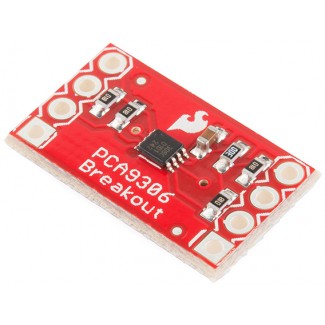
**Schéma du module**



**Test alimentation en 5V  = DESTRUCTION DU µC et du capteur !!!!!**

* **adapter en 3,3V avec un module BSS138 ou un PCA9306 + régulateur 3,3V**



****

**Pour une Arduino Uno :**

* Exemple de programme : TMP102.ino

**Pour la carte Panda2 :**

* Exemple de programme : TMP102\_Tester.sln

**Etalonnage** avec un MX59HD : (PT100 ou PT1000) selon le principe de la fiche d’étalonnage

**Ressources documentaires :**

Informations sur le bus I2C : cours et TD SSI

<http://www.equinoxefr.org/post/2012/11/26/conversion-de-signaux-i2c-3-3v-5v/>

Datasheet : TMP102

Module SEN-09418 TMP102 (I2C) :

<https://www.sparkfun.com/products/11931>

Manuel d’utilisation du multimètre MX59HD

PT100, PT1000 :

<http://fr.wikipedia.org/wiki/Thermom%C3%A8tre_%C3%A0_r%C3%A9sistance_de_platine>

Exemple de programme pour un Arduino Uno

//Arduino 1.0+ Only

//Arduino 1.0+ Only

//////////////////////////////////////////////////////////////////

//©2011 bildr

//Released under the MIT License - Please reuse change and share

//Simple code for the TMP102, simply prints temperature via serial

//////////////////////////////////////////////////////////////////

#include <Wire.h>

int tmp102Address = 0x48;

void setup(){

Serial.begin(9600);

Wire.begin();

}

void loop(){

float celsius = getTemperature();

Serial.print("Celsius: ");

Serial.println(celsius);

float fahrenheit = (1.8 \* celsius) + 32;

Serial.print("Fahrenheit: ");

Serial.println(fahrenheit);

delay(1000); //just here to slow down the output. You can remove this

}

float getTemperature(){

Wire.requestFrom(tmp102Address,2);

byte MSB = Wire.read();

byte LSB = Wire.read();

//it's a 12bit int, using two's compliment for negative

int TemperatureSum = ((MSB << 8) | LSB) >> 4;

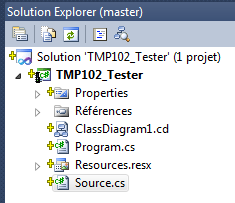
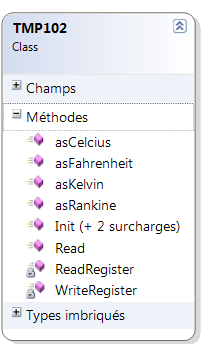
float celsius = TemperatureSum\*0.0625;

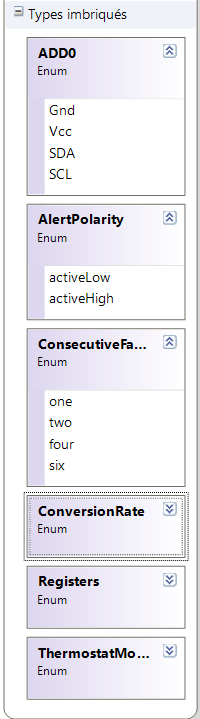
return celsius;

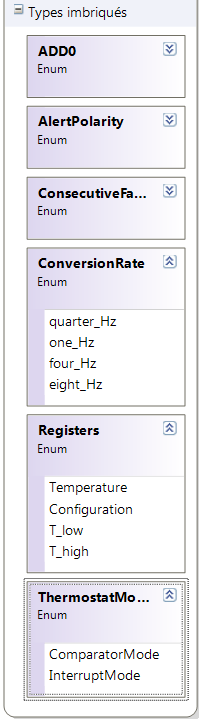
}

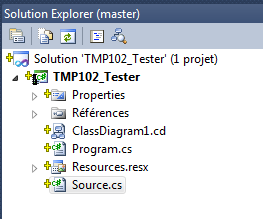
Exemple de programme pour un PANDA 2

using System;









using System.Threading;

using Microsoft.SPOT;

using Microsoft.SPOT.Hardware;

using GHIElectronics.NETMF.FEZ;

using Sensor;

namespace TMP102\_Tester

{

public class Program

{

public static void Main()

{

// Disable garbage collector messages

Debug.EnableGCMessages(false);

bool ledState = false;

OutputPort led = new OutputPort((Cpu.Pin)FEZ\_Pin.Digital.LED, ledState);

TMP102 CPTTEMP102 = new TMP102();

CPTTEMP102.Init(TMP102.ADD0.Gnd);

while (true)

{

CPTTEMP102.Read();

Debug.Print("Temperature: " + CPTTEMP102.asCelcius() + " C");

//Debug.Print("Temperature: " + CPTTEMP102.asFahrenheit() + " F");

// Sleep for 500 milliseconds

Thread.Sleep(500);

// toggle LED state

ledState = !ledState;

led.Write(ledState);

}

}

}

}