

La PT1000

Les sondes de température de type platine (PRTD - Platinum Resistance Temperature Detector) sont des capteurs d'un emploi courant dans l'industrie, leur principe repose sur la variation de résistance en température d'un conducteur de platine (fil ou feuille) selon des caractéristiques électriques connues et stables, décrites par les normes IEC 751, DIN 43760...



Les modèles rencontrés sont référencés Pt100, Pt500, Pt1000 : la dénomination du modèle le plus courant Pt100 signifie que le capteur présente une résistance de 100 ohms à 0°C. Elles peuvent être utilisées dans des gammes de températures allant au-delà de -200°C à 800°C. La variation de résistance est quasiment linéaire en fonction de la température ; elle est pour une Pt100 de l'ordre de +0.4 ohm/°C à 0°C et de +0.35 ohm/°C à 300°C.

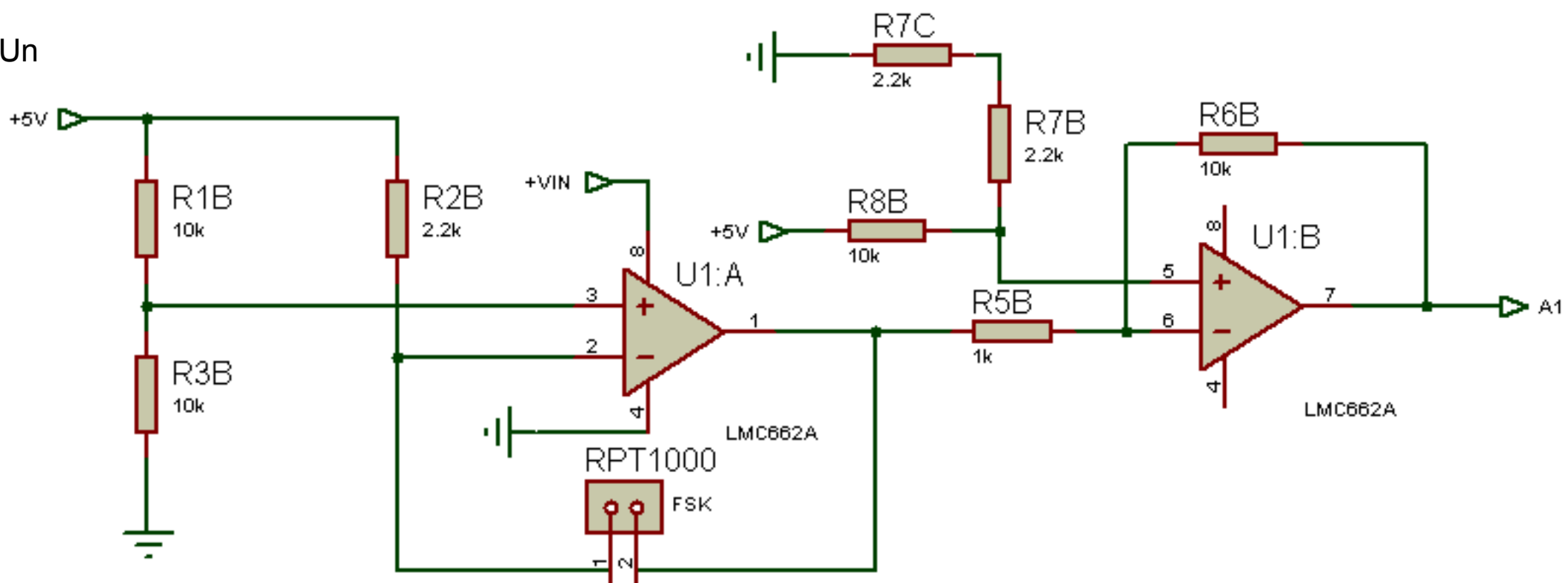
Dans notre cas l'utilisation d'une Pt1000 permet de négliger les résistances de contact et d'utiliser un montage à deux fils.

$$R_T = R_0 \cdot (1 + aT)$$
 avec $a = 3.85 \cdot 10^{-3} \text{ }^{\circ}\text{C}^{-1}$ et $R_0 = 1000 \text{ Ohms}$

Le conditionneur-transmetteur

Cahier des charges :

- Un



- Arduino ne peut mesurer qu'une tension comprise entre 0 et 5 V.
- Notre étendue de mesure souhaitée est de -50°C à 40°C
- Par conséquent on souhaite obtenir -50°C -> 0,5 V et 40°C -> 4,5V
- Pour éviter l'autoéchauffement : $I_{PT1000} < 1,2 \text{ mA}$
- Alimentation disponible 0V - 9V (pas d'alimentation symétrique)
- Choisir un AmpliOp supportant une alimentation non symétrique (Single Supply, éviter le classique TL081...)LM324, LM358 : $V_{\text{output}} = 0 \text{ à } V_{CC} - 1,5$
LMC662 : Rail to Rail, $V_{\text{output}} = 0 - 0,2 \text{ à } V_{CC} - 0,2$

Etage 1 : le générateur de courant

Point de repos : $V^+ = 10/(10+10) \cdot 5 = 2,5 \text{ V}$

Il s'agit d'un montage linéaire donc $V_d = 0 \text{ V}$

Loi des noeuds ($I^+ = I^- = 0$) : $I_{R2B} = I_{PT1000}$

D'après la loi d'ohm : $I = I_{PT1000} = (5 - V^-)/R_{2B} = (5-2.5)/2200 = 1,136 \text{ mA}$

D'après la loi des mailles : $V_{S1} + V_{PT1000} - V^- = 0$

$$\Leftrightarrow V_{S1} = V^- - R_{PT1000} \cdot I = 2,5 - 1000(1+aT) \cdot 1,136 \cdot 10^{-3} = 2,5 - 1,136 - 1.136 \cdot 0.00385 \cdot T = 1,364 - 4,37 \cdot 10^{-3} T$$

Si T = -50°C => $V_{S1}(-50^{\circ}\text{C}) = 1,5825\text{ V}$

Si T = 40 °C => $V_{S1}(40^{\circ}\text{C}) = 1,1892\text{ V}$

Sensibilité du montage : $S_1 = dV/dT = 4,37\text{ mV/}^{\circ}\text{C}$

Etage 2 : L'amplificateur inverseur

D'après le diviseur de tension : $V^+ = (2,2+2,2)/(10+2.2+2.2) * 5 = 1,5278\text{ V}$

Il s'agit d'un montage linéaire donc $V^+ = V^-$, d'après le théorème de Millman :

$V^- = (R_{6B} * V_{S1} + R_{5B} * V_{S2}) / (R_{6B} + R_{5B})$

$V_{S2} = ((R_{6B} + R_{5B}) * V^- - R_{6B} V_{S1}) / R_{5B} = -10 * V_{S1} + 16,8058$

$V_{S2} = -10 (1,364 - 4,37 \cdot 10^{-3} T) + 16.8058$

$V_{S2} = 43,7 \cdot 10^{-3} * T + 3,1658$

Si T = - 50°C => $V_{S2}(-50^{\circ}\text{C}) = 0,98\text{ V}$

Si T = 40 °C => $V_{S1}(40^{\circ}\text{C}) = 4,91\text{ V}$

Sensibilité du montage : $S_2 = dV/dT = 43,7\text{ mV/}^{\circ}\text{C}$

Nous avons choisi de ne pas descendre trop bas $V_{S2}(-50^{\circ}\text{C}) = 0,98\text{ V}$, car notre alimentation non symétrique nous permet pas d'atteindre 0 V

Le choix d'un Ampli Op "Rail to Rail" nous permet d'obtenir un V_{sat} proche de l'alimentation.
D'après le datasheet du LMC662 : $V_{sat-} = V^- + 0,2 = 0,2\text{ V}$ et $V_{sat+} = V^+ - 0,2 = 9,3\text{ V}$ (pour un AmpliOp classique : $V_{sat+} = V^+ - 1,5$)

Ce montage répond au cahier des charges.

Cette fonction de transfert est théorique : la dispersion des composants et l'offset de l'ampliOp rendent cette formule approximative.

Il faut maintenant étalonner ce conditionneur.

Etalonnage final

Voici la courbe d'étalonnage obtenue avec la carte finale :

En abscisse : tension numérisée par le CAN de l'Arduino (1024 <-> 5 V)

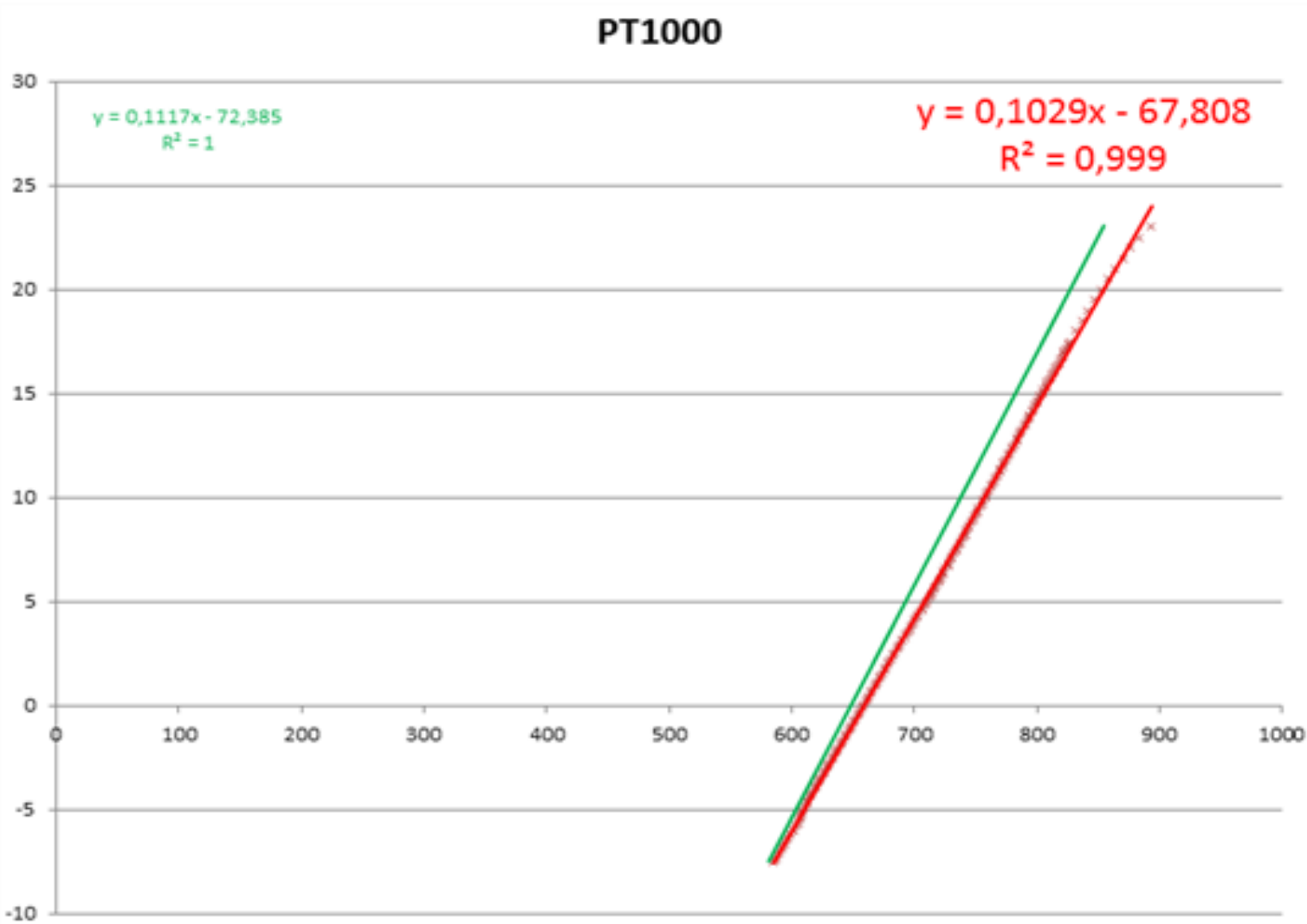
La courbe verte est la courbe théorique établie ci-dessus.

Les points de mesure sont en rouge.

La courbe d'étalonnage a un excellent coefficient de corrélation $r^2=0,9996$.

La différence entre la courbe théorique et la courbe expérimentale s'explique par :

- la dispersion des composants (résistances à 1 % - Erreur de gain et d'offset)
- l'offset des AmpliOp (Erreur d'offset)
- l'exactitude de notre thermomètre étalon qui était un DHT22
- les conditions de l'expérience : mesure à la volée dans une boîte isotherme (pour limiter [l'erreur de traînage](#)) à l'intérieur d'un congélateur.



Les erreurs d'offset et de gain sont corrigées par notre étalonnage. Le coefficient de corrélation élevé nous permet d'avoir confiance dans notre équation :

```
tPT= analogRead(PT1000);  
tPT=0.1029*tPT-67.808;    // Etalonnage par DHT22
```

Attention : ce montage est sensible aux perturbations électromagnétiques de l'émetteur Kiwi.
Il faut veiller à éloigner ce montage de l'émetteur !

Test et étalonnage

`/* Objectif`

`Lire la tension de sortie du montage 1 Pile A0 -> M0`

`Lire la tension de sortie du montage 2 PT1000 A1 -> M1`

`Lire la tension de sortie du montage 3 CTN 10k A2 -> M2`

`Lire la température indiquée par le dht22`

`*/`

`#include <DHT.h>`

`#include <Wire.h>`

`#include <LiquidCrystal_I2C.h>`

`#include <SD.h>`

`#define DHTPIN 3 // what pin we're connected to`

`#define DHTTYPE DHT22 // DHT 22 (AM2302)`

`LiquidCrystal_I2C lcd(0x27,20,4);`

`DHT dht(DHTPIN, DHTTYPE);`

`int mes,i,M0=0,M1,M2;`

`int chipSelect=10;`

`int indicatorLed = 13;`

`boolean error=false;`

`char message[140];`

`float t,VPT1000,VCTN,Pile;`

`void setup()`

`{`

`Serial.begin(9600);`

`pinMode(indicatorLed, OUTPUT);`

`lcd.init(); // initialize the lcd`

`lcd.noBacklight();`

`dht.begin(); // Initialiser le capteur dht22`

`lcd.setCursor(15, 0);`

`if (!SD.begin(chipSelect)) {error=true;lcd.print("SD Er");} // Initialisation carte SD`

`else {lcd.print("SD OK");iniSD();}`

`lcd.setCursor(0, 0);`

`lcd.print("M");`

`lcd.setCursor(7, 0);`

`lcd.print("T=");`

`lcd.setCursor(0, 1);`

`lcd.print("Mes1 PT100 =");`

`lcd.setCursor(0, 2);`

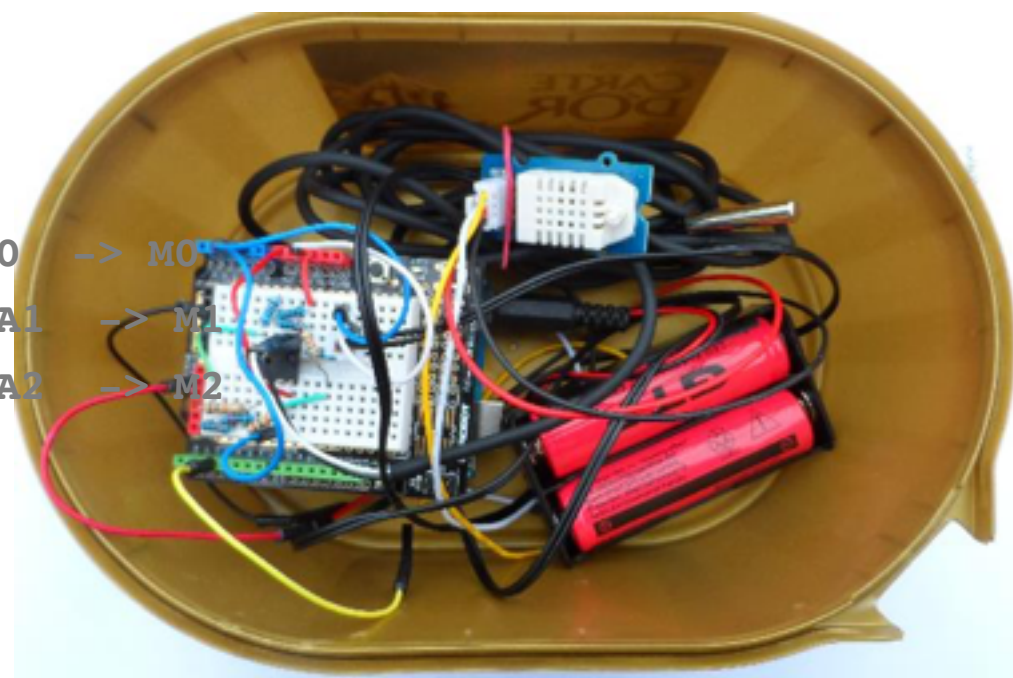
`lcd.print("Mes2 PT1000 =");`

`lcd.setCursor(0, 3);`

`lcd.print("Mes3 CTN =");`

`}`

`void loop()`




```

{
    if(i%30==3) lcd.noBacklight();
    t = dht.readTemperature();
    M0=analogRead(A0);
    M1=analogRead(A1);
    M2=analogRead(A2);
    VPT1000=5.0/1024.0*M1;
    VCTN=5.0/1024.0*M2;
    Pile=10.0/1024.0*M0;
    affiche();
    if (i%30==0) ecrireSD();
    i++;
    delay(1000); // Show new results every second.
}

void affiche() {
    lcd.setCursor(9, 0);
    lcd.print(t,1);
    lcd.setCursor(14, 1);
    lcd.print(M0);
    lcd.print(" ");
    lcd.setCursor(14, 2);
    lcd.print(M1);
    lcd.print(" ");
    lcd.setCursor(14, 3);
    lcd.print(M2);
    lcd.print(" ");
    Serial.print(t,1);Serial.print(" C\t Pile=");
    Serial.print(Pile,1);Serial.print(" V\t CTN=");
    Serial.print(VCTN,3);Serial.print(" V\t PT1000=");
    Serial.print(VPT1000,3);Serial.println(" V");
    if (error) Serial.println ("Erreur SD");
}

void iniSD() {
    File TEMPtab = SD.open("TEMPtab.csv", FILE_WRITE);
    if (TEMPtab) {
        error=false;
        TEMPtab.println("mes;t(dht);Pile;PT1000;CTN;VCTN;VPT1000");
        TEMPtab.close();
    }
    else {
        error=true;
        lcd.setCursor(15,0);
        lcd.print("SD Er");}
}

void ecrireSD(){
    mes++;
    lcd.backlight();
    lcd.setCursor(1, 0);
    lcd.print(mes);
    File TEMPtab = SD.open("TEMPtab.csv", FILE_WRITE);
    if (TEMPtab) {
        error=false;

```

```
    TEMPtab.print(mes);
    TEMPtab.print(";");
    TEMPtab.print(t,1);
    TEMPtab.print(";");
    TEMPtab.print(Pile,1);
    TEMPtab.print(";");
    TEMPtab.print(M1);
    TEMPtab.print(";");
    TEMPtab.print(M2);
    TEMPtab.print(";");
    TEMPtab.print(VCTN,3);
    TEMPtab.print(";");
    TEMPtab.println(VPT1000,3);
    TEMPtab.close();
}
else {
    error=true;
    lcd.setCursor(15,0);
    lcd.print("SD Er");}
}
```