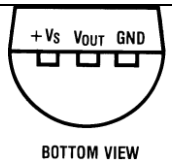
Fiche Pratik

001	**Titre : Mesure de température interfacée par carte Arduino**
Type de réalisation : *montage électronique, de surveillance de température*	**Concepteur :** C. Rouviere **Coordonnées** : Laboratoire lbv villefranche/mer **Courriel :** rouviere@obs-vlfr.fr
Durée estimée :1 journée	**Date de la réalisation : 2011**
Fichiers associés *(Plans mécaniques, Schémas électroniques…)* **:**	

Objectif : Capture de la température d'une platine de microscope et génération d'un fichier Log ou s'inscrit le temps ordinateur et la température relevée.

Matériel *(Liste/Références/Fournisseurs/Prix unitaires et coût global du matériel nécessaire)*

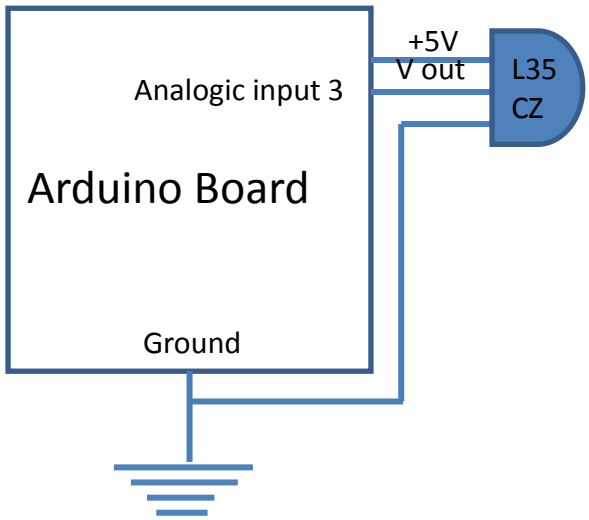
Description		Fournisseur	Référence	Nombre	PU HT
			LM35CZ, LM35AZ LM35DZ	1	5/6 euros

Réalisation

1. Description

Le dispositif consiste à positionner un capteur de température (lm35DT) sur une platine de microscope. Il est connecté à un Arduino. Son rôle est d'acquérir régulièrement la température locale et de l'écrire dans un fichier texte avec l'heure du relevé. La température est simultanément affichée sur l'écran de l'ordinateur

2. Schéma fonctionnel



3. Interconnexions

4. Software/Firmware

a. Firmware

Voici le code à télécharger dans l'Arduino :

```
////////////////////////////////////////////////////////////////////
/*
   CapteurTemperature
   http://maths-sciences.net
*/
////////////////////////////////////////////////////////////////////

int analogPin = 0; // Patte centrale LM35 sur l'entrée analogique 3.
int value = 0;     // Variable de stockage de la valeur mesurée.
float tempC;

void setup()
{
  Serial.begin(9600); // Transfert des données série a 9600 bauds.
}

void loop()
{
  value = analogRead(analogPin);  // lecture en entrée de la pin 3.
  tempC=(5.0*value*100.0)/1024;   //conversion de la valeur
analogique temperature
  Serial.println(tempC);          // Debug value.
  delay(100);                     // attente de 100 millisec.
}
```

b. Software

Voici le code en Processing à exporter en exécutable (.exe)

```
////////////////////////////////////////////////////////////////////
/*
   CapteurTemperature
   http://maths-sciences.net
*/
////////////////////////////////////////////////////////////////////

import processing.serial.*;  // Charge la bibliothèque serial.

Serial myPort;          //Création de l'objet myPort (classe Serial).
int baudrate = 9600;  //Vitesse de transfert des données (en bauds).
int valPort = 0;       // Données reçues depuis le port série.
String buffer = "";
// Tampon pour récupérer la dernière valeur mesurée sous la forme
//d'une chaine de caractères. Valeur mesurée par la carte Arduino
//codée sur 10 bits (entre 0 et 1023 en décimal)
int value = 0;

// Tensions électriques
```

```
float tension = 0;
float tempC=0;
float tensionMax = 5.0;
// La tension entre la borne 5 V et Gnd est sur ma carte Arduino de
//4,8 V si l'alimentation se fait par le port USB, sinon elle est de
//5,0 V si l'alimentation se fait par la prise jack 2,1 mm. Modifier
//cette valeur en conséquence.
// Température
float temperature = 0;   // La température est en degrés Celsius.
PrintWriter output; // Création d'un objet PrintWriter
// Chargement des polices
PFont policeT;
PFont policem;
int fps = 1;               // Nombre de frames par seconde

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

void setup()
{
  frameRate(fps);
  size(250, 100);
  policeT = loadFont("CourierNewPS-BoldMT-48.vlw");
  policem = loadFont("CourierNewPS-BoldMT-12.vlw");

  println("Ports séries disponibles :");
  println(Serial.list());

// Sur mon ordinateur, la carte Arduino est connectée au port COM3,
// le deuxième dans la liste, le 1 dans Serial.list()[1].
  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, baudrate);

  // Créer un fichier données.txt dans le répertoire sketch
  output = createWriter("données.txt");
}

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

void draw()
{
  background(0);
  stroke(255);

  while (myPort.available() > 0) {
//  Pour transmettre la valeur mesurée codée sur 10 bits (2^10 =
//1024),soit un nombre compris entre 0 et 1023, valPort prend
//successivement des valeurs entre 48 et 57, ce qui correspond en
//code ASCII aux caractères 0 à 9. Quand la valeur à transmettre (0
//à 1023) l'est, valPort prend les valeurs 13 (retour chariot en
//code ASCII), puis 10 (saut à la ligne en code ASCII).
    valPort = myPort.read();
    serialEvent(valPort);
  }
// Convertit value en une tension en volts (un simple produit en
//croix). tension = tensionMax * value / 1024;
```

```
// L'amplification avec l'ampli op non inverseur est de Vs/Ve = (1 +
//R3 / R2)
// R3 théorique  = 33 kilohms
// R2 théorique = 4,7 kilohms
// Amplification théorique = (1 + (33 / 4.7)) = 8,0
// A un instant donné en sortie du capteur on a 0,21 V
// Au niveau de l'entrée analogique de la carte Arduino on a 1,73V.
// L'amplification mesurée est donc de 1,73 / 0,21 = 8,2.
// tension=tension / 8.2; On calcule la tension avant amplification.
// On convertit cette tension en une température (un simple produit
//en croix). temperature = tension / 0.01; Le capteur de température
//LM35CZ fournit une réponse linéaire de 10 mV (0,01 V) par °C
// Dans l'intervalle [+ 2 °C ; + 110°C].
// Affichage de la température à l'Ecran (à chaque boucle, toutes
//les secondes : cf fps).
  if (tempC != 0) {
    int s = second();  // Values from 0 - 59
    int m = minute();  // Values from 0 - 59
    int h = hour();    // Values from 0 - 23

// PFont police = loadFont("CourierNewPS-BoldMT-48.vlw");
    textFont(policeT, 48);
    textAlign(CENTER, CENTER);
    text(nf(tempC, 2, 1) + "Â°C", width/2, height/2);
    output.println(h+ "h "+ m + "m "+ s + "s " + "\t" +nf(tempC, 2,
1)+ "Â°C");

    textFont(policem, 12);
    textAlign(LEFT, BOTTOM);
    text("Press ESC to stop...@C.Rouviere", 0, height);
  }
}
//////////////////////////////////////////////////////////////////////

void serialEvent(int serial)  // Méthode de la classe Serial.
{
  if (serial != 10) {      // 10 <=> saut Ã  la ligne en code ASCII.
    buffer += char(serial); // Store all the characters on the line.
    }
  else {
// The end of each line is marked by two characters, a carriage
//return (13) and a newline (10). We're here because we've gotten a
//newline, but we still need to strip off the carriage return.
    buffer = buffer.substring(0, buffer.length()-1);
//Parse the String into an integer (analog inputs go from 0 to
1023).
    tempC = parseFloat(buffer);
    println(tempC);
    // Clear the value of "buffer"
    buffer = "";
  }
}

//////////////////////////////////////////////////////////////////////
```

```
void keyPressed() {
  if (key == ESC) {
    output.flush();  // Writes the remaining data to the file
    output.close();  // Finishes the file
    exit();          // Stops the program
  }
}
```

5. Typon

6. Mode d'emploi

Après son lancement la température va s'afficher sur l'écran et quand on arête l'acquisition, (il suffit de presser la touche ESC) Les données Temps + Températures sont écrites dans un fichier données.txt dans le répertoire application 32 bits.

Voici un exemple de fichier données.txt :

donnée.txt :

11h 54m 31s	54,7°C
11h 54m 32s	54,7°C
11h 54m 33s	54,7°C
11h 54m 34s	54,7°C
11h 54m 35s	54,7°C
11h 54m 36s	54,7°C
11h 54m 37s	54,7°C
11h 54m 38s	54,7°C
11h 54m 39s	54,7°C
11h 54m 40s	54,7°C
11h 54m 41s	54,7°C
11h 54m 42s	54,7°C
11h 54m 43s	54,7°C
11h 54m 44s	54,7°C
11h 54m 45s	54,7°C
11h 54m 46s	54,7°C
11h 54m 47s	54,7°C
11h 54m 48s	54,2°C
11h 54m 49s	54,2°C
11h 54m 50s	54,2°C
11h 54m 51s	54,2°C
11h 54m 52s	54,2°C
11h 54m 53s	54,2°C
11h 54m 54s	54,2°C
11h 54m 55s	54,2°C

7 Remarques

Utiliser un câble de données blindé (de type coaxial) si vous désirez déporter le capteur.
Tarif 65 euros/100 m !