

Info 2 PC

TP 3

Au cours de ce TP, nous allons utiliser une thermistance pour acquérir des mesures de température, les enregistrer à l'aide de fichiers, puis nous interpréterons ces données à l'aide de tableurs (Calc par exemple, **gnuplot** pour les adeptes du terminal).

Vous pourrez retrouver un exemple de manipulation de la librairie SD page 108 dans le dernier cours. (<http://www-perso.unilim.fr/deneuille/docs/Info2PC/Cours4.pdf>).

Vous pourrez trouver toute la doc nécessaire à ce TP sur les pages 110 et suivantes de ce même cours.

Commencez par vous rendre à l'endroit où est installé l'environnement de développement :

```
cd /home/shared/arduino/arduino-1.6.0/
```

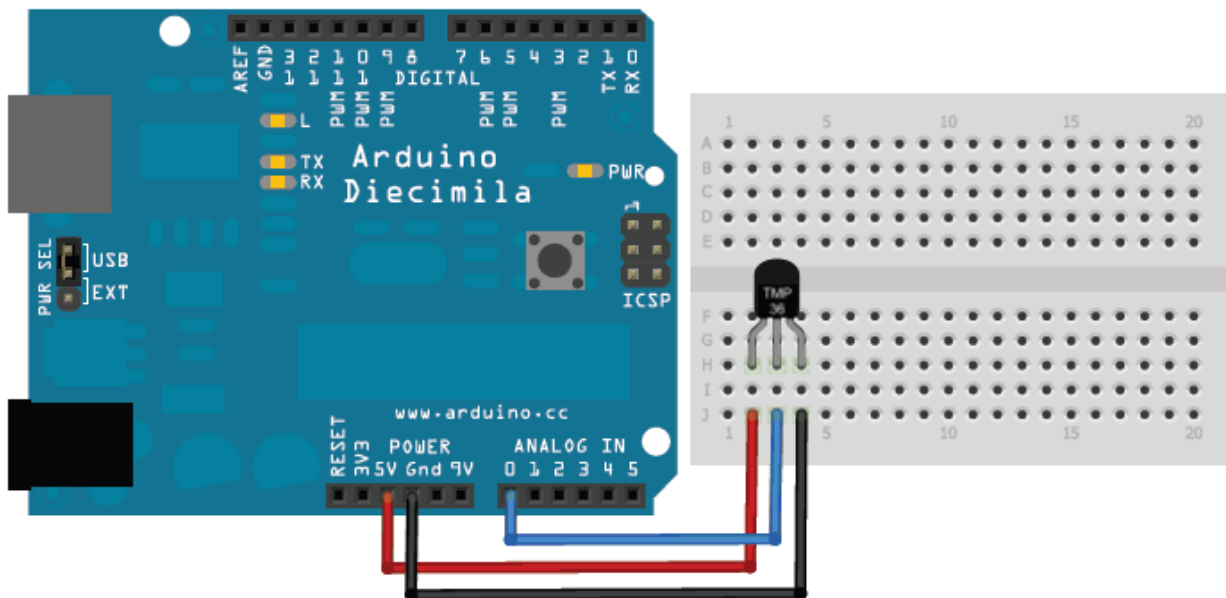
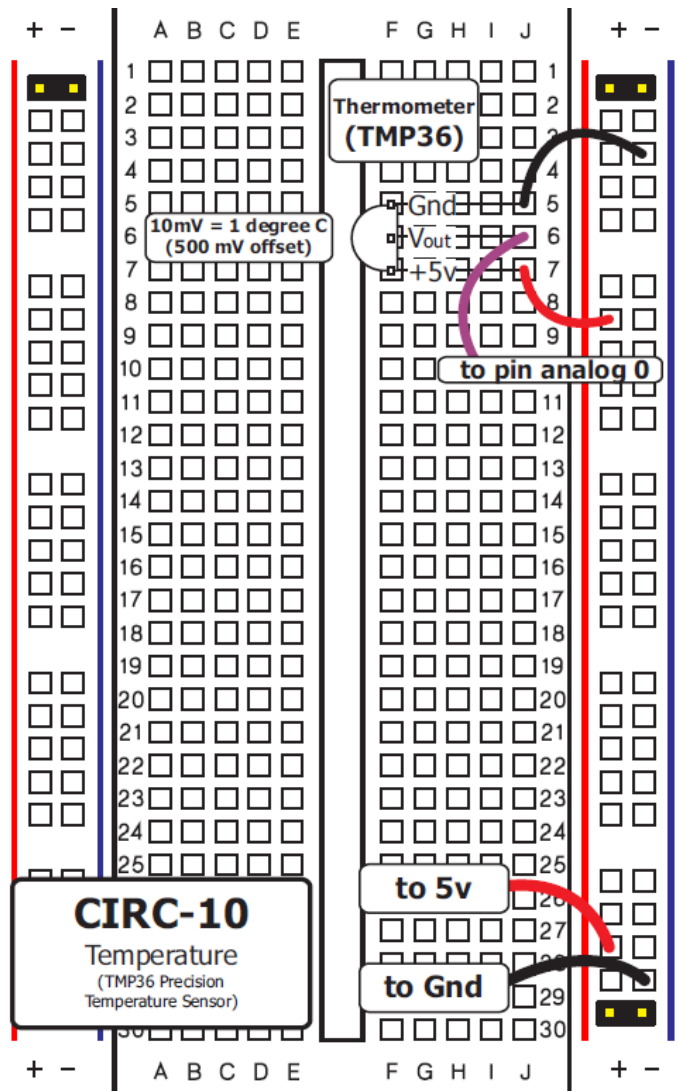
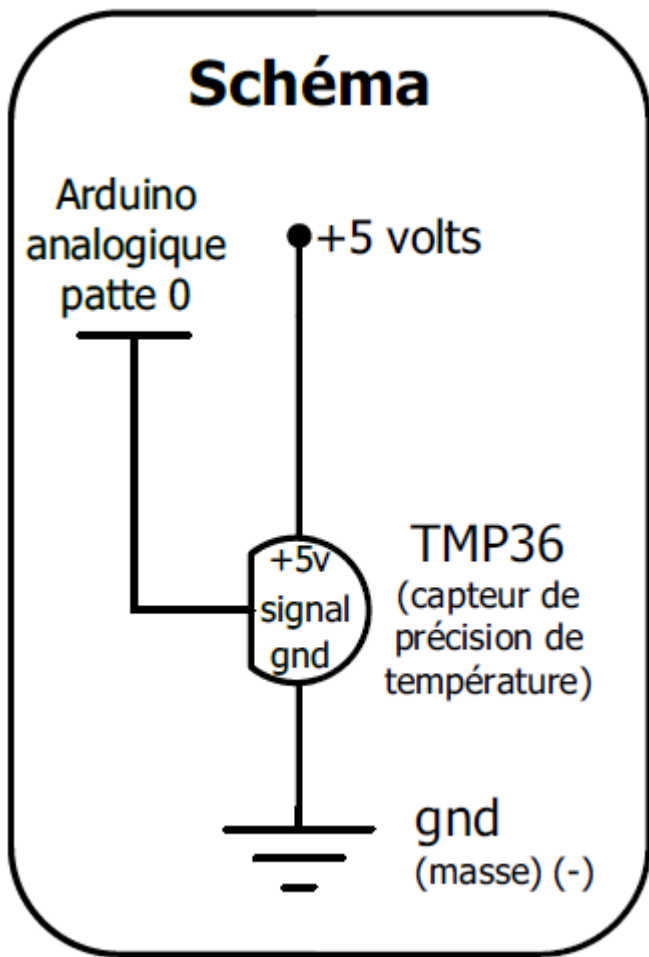
Puis lancez l'environnement de développement :

```
./arduino &
```

ATTENTION

Relier la patte 5V du Arduino à la masse peut le tuer, même si c'est fait via la breadboard.

Commencez par réaliser le montage de la page suivante.



Voici le code à téléverser (puis ouvrir le moniteur série) :

```
/* -----
 * | Code d'Exemple du Kit d'Expérience Arduino |
 * | CIRC-10 .: Température :. |
 * -----
 * Un programme simple pour afficher la température
 * sur la console du logiciel
 * Pour plus de détails sur ce circuit :
 */

//assignement des pattes du TMP36
int temperaturePin = 0; //la patte analogique à laquelle la patte Vout du TMP36 est connectée
                        //la resolution est de 10 mV / degré Celsius
                        //(500 mV d'offset) pour obtenir des températures négatives

void setup()
{
  Serial.begin(9600); //lance la connexion série avec l'ordinateur
                    //pour voir le résultat ouvrez la console série
                    //dernier bouton sous la barre de menu
                    //(il ressemble à rectangle avec une antenne)
}

void loop() //s'exécute à l'infini
{
  float temperature = getVoltage(temperaturePin); //obtiens la tension de sortie du
                                                  //capteur de temperature

  temperature = (temperature - .5) * 100; //conversion de 10mV
                                          //par degre avec 500mV d'offset en
                                          //degrés ((tension - 500mV) * 100)

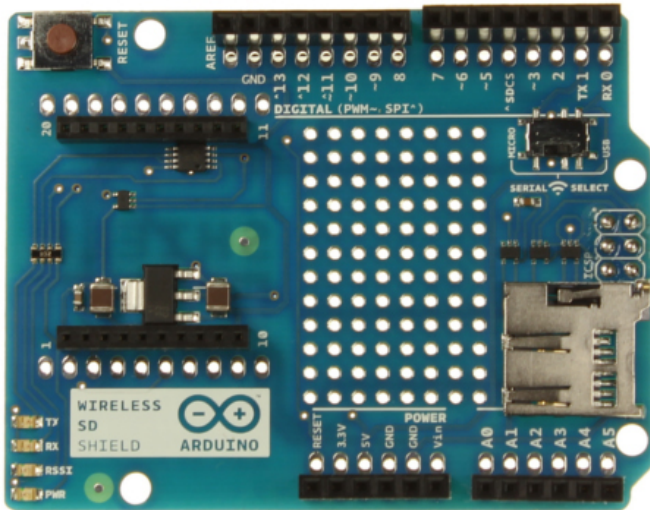
  Serial.println(temperature); //affiche le résultat
  delay(1000); //attend une seconde
}

/*
 * getVoltage(int pin) - retourne la tension sur la patte analogique
 * définie par pin
 */
float getVoltage(int pin){
  return (analogRead(pin) * .004882814); //Convertie d'une plage de 0 à 1023
                                          //à une plage de 0 à 5V (Chaque pas égal environ 5 mV)
}
}
```

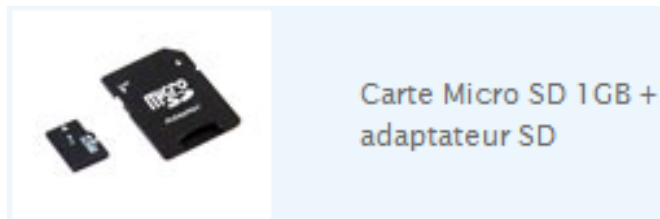
Pour améliorer le montage vous pouvez changer le `Serial.begin(9600)` par `Serial.begin(115200)` afin de communiquer 12 fois plus vite avec le PC !

Enfichez moi (la paix)

Passons maintenant à l'ajout du shield pour la carte SD (merci d'y aller en douceur)



Placer la carte microSD à l'intérieur (sans l'adaptateur !)



Rebranchez les fils sur les bonnes pattes.

Téléversez le code suivant puis ouvrez le moniteur série :

```
#include <SD.h>
const int chipSelect = 4;

//assignement des pattes du TMP36
int temperaturePin = 0; //la patte analogique à laquelle la patte Vout du TMP36 est connectée
                        //la resolution est de 10 mV / degré Celsius
                        //(500 mV d'offset) pour obtenir des températures négatives

long timeRef=0; // variable pour temps de référence en millisecondes

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized.");
  timeRef=millis(); // initialisation de la référence du comptage
}

void loop()
{
  // make a string for assembling the data to log:
  String dataString = "";

  dataString += String(millis()-timeRef);
  dataString += ",";
  dataString += String(analogRead(temperaturePin));

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  File dataFile = SD.open("datalog.csv", FILE_WRITE);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }

  delay(1000); //attend une seconde
}
```

Faites varier un peu la température (avec la glace, en touchant le capteur, etc.)

Au bout de 5 minutes, récupérez la carte microSD, mettez la dans son adaptateur et branchez sur le port USB de l'ordinateur le lecteur de carte SD qu'on vous a fourni.

Insérez la carte et récupérez le fichier « datalog.csv » sur votre compte. Ouvrez « Calc »

(LibreOffice/OpenOffice) et faites un diagramme illustrant les modifications de températures en fonction du temps en n'oubliant pas avant de réaliser au préalable les conversions :

$=((VAL_ANAL * 0,004882814) - 0,5) * 100$