LYCÉE de l'ÉLORN
le choix de votre réussite

**Sciences de l'Ingénieur**

# Intégration d'un capteur analogique sur un système numérique

## 1- Objectif de l'activité

Vous allez étudier l'intégration d'un capteur analogique sur un système numérique. Après avoir validé le fonctionnement de ce capteur sur une interface Arduino vous utiliserez un Shield Internet pour proposer un accès à la donnée prélevée.

### 1.1- Cahier des charges.

On souhaite réaliser un système de contrôle de température à partir d'un système **Arduino**. Ce dispositif devra être en mesure de transmettre sur une page WEB la température d'un local au degré prêt sur une plage de 10 à 40°C.

## 2- Travail demandé

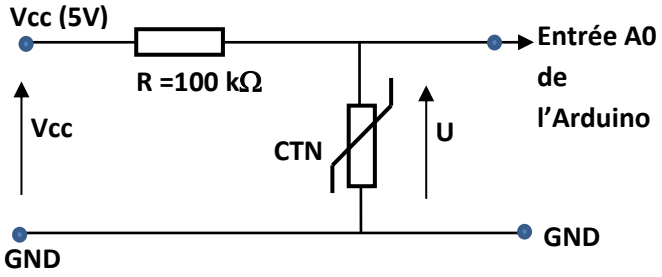
### 2.1- Utilisation d'une thermistance CTN

La CTN (coefficient de température négatif) est un capteur de température passif. Sa résistance varie en fonction de la température : elle diminue de façon uniforme lorsque la température augmente, et inversement. On suppose que ce composant a une résistance nominale de **100 kΩ à 26 °C**.

**2.1.1** Mesurer, à la température ambiante $T_0$ **= 20 °C**, la résistance de la CTN qui vous a été donnée.

**2.1.2** Relever sur la caractéristique ***R = f(T°)*** du document DT1 la valeur Ohmique du composant aux températures $T_1$ **= 10 °C** et $T_2$ **= 40 °C**.

La CTN sera utilisée dans un câblage de deux résistances, la tension à ses bornes sera directement injectée sur le convertisseur analogique numérique de l'Arduino.




**2.1.3** Dans ce montage, **calculer** la valeur de la tension vue aux bornes de la CTN (tension U) pour les trois températures précédentes.

**2.1.4** La tension U est destinée à être numérisée par le CAN de l'Arduino. **Donner** l'équation de conversion d'après les éléments vus dans sur les DT2 et DT3. **Calculer** les grandeurs numériques N de sortie du convertisseur attendues pour $T_0$, $T_1$ et $T_1$.

**2.1.5 En déduire** la fonction capable de calculer la température à partir des valeurs numériques N si l'on considère que le comportement de la CTN est linéaire sur cette plage de valeur.

## 2.2- Modélisation du conditionneur sous Matlab

Le fichier de simulation « **CTN_Ardiuno.mdl** » vous permettra de vérifier les valeurs précédentes et de préparer le programme informatique demandé.

**2.2.1** Après avoir **saisi la partie CAN du modèle Matlab** donner les valeurs numériques issues du convertisseur CAN pour les trois seuils de températures exploités.

**2.2.2 Ajouter** au modèle la structure permettant d'afficher la valeur résistive de la CTN.

**2.2.2 Ajouter** au modèle la structure permettant d'afficher la valeur de la température à partir de la valeur numérique N.

## 2.3- Mise en œuvre

**2.3.1** Réaliser le câblage du capteur sur l'Arduino puis saisir le programme suivant sur l'IDE.

```
int conversion=0;
float tension=0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  conversion = analogRead(A0);
  tension = 5*float(conversion)/1023;
  Serial.print("valeur numerique prelevee= " );
  Serial.println(conversion);
  Serial.print("Tension (en Volt)= " );
  Serial.println(tension);
  delay(1000);
}
```

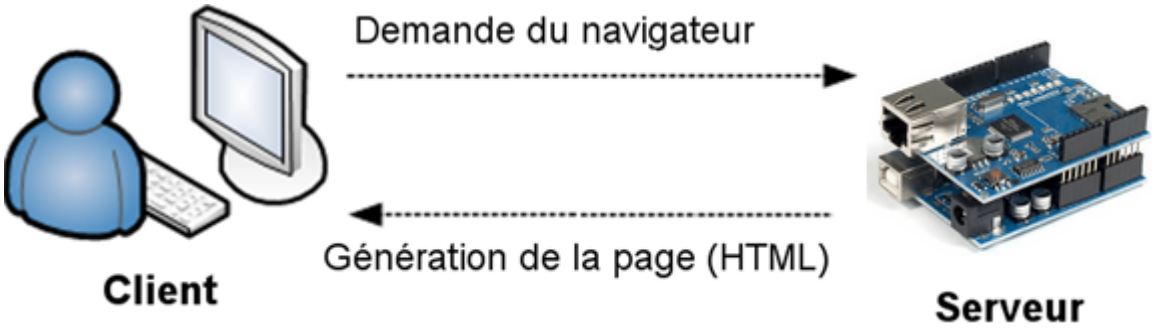
**2.3.2 Exécuter** le programme, lancer le moniteur série (Ctrl+Maj+M) pour constater le fonctionnement de l'ensemble.

**2.3.3 Estimer** le temps de réponse du capteur face à un échelon de température puis comparer cette valeur à l'aide du modèle multiphysique.

**2.3.4 Modifier** le programme afin d'afficher la température mesurée en plus de la tension U.

## 2- Création d'un serveur WEB à partir d'un Arduino et d'un Shield Ethernet.

En mode serveur, c'est l'Arduino qui doit générer le site web dès qu'un autre système s'adressera à lui.




**2.1** Vous allez reprendre le paramétrage réseau de votre ordinateur pour le transférer à l'Arduino.

Sur un PC de la salle exécuter la commande IPCONFIG / ALL sur une invite de commande puis relever :

- l'adresse IP de votre ordinateur
- le masque de sous-réseau
- l'adresse MAC
- l'adresse IP de la passerelle

Ajouter un Shield Internet à votre Arduino (décâbler puis recâbler le capteur)

**2.2 Ouvrir** le fichier WebServeur_DEMO.ino puis compléter les paramètres MAC, IP, masque et passerelle prélevés en 2.1.

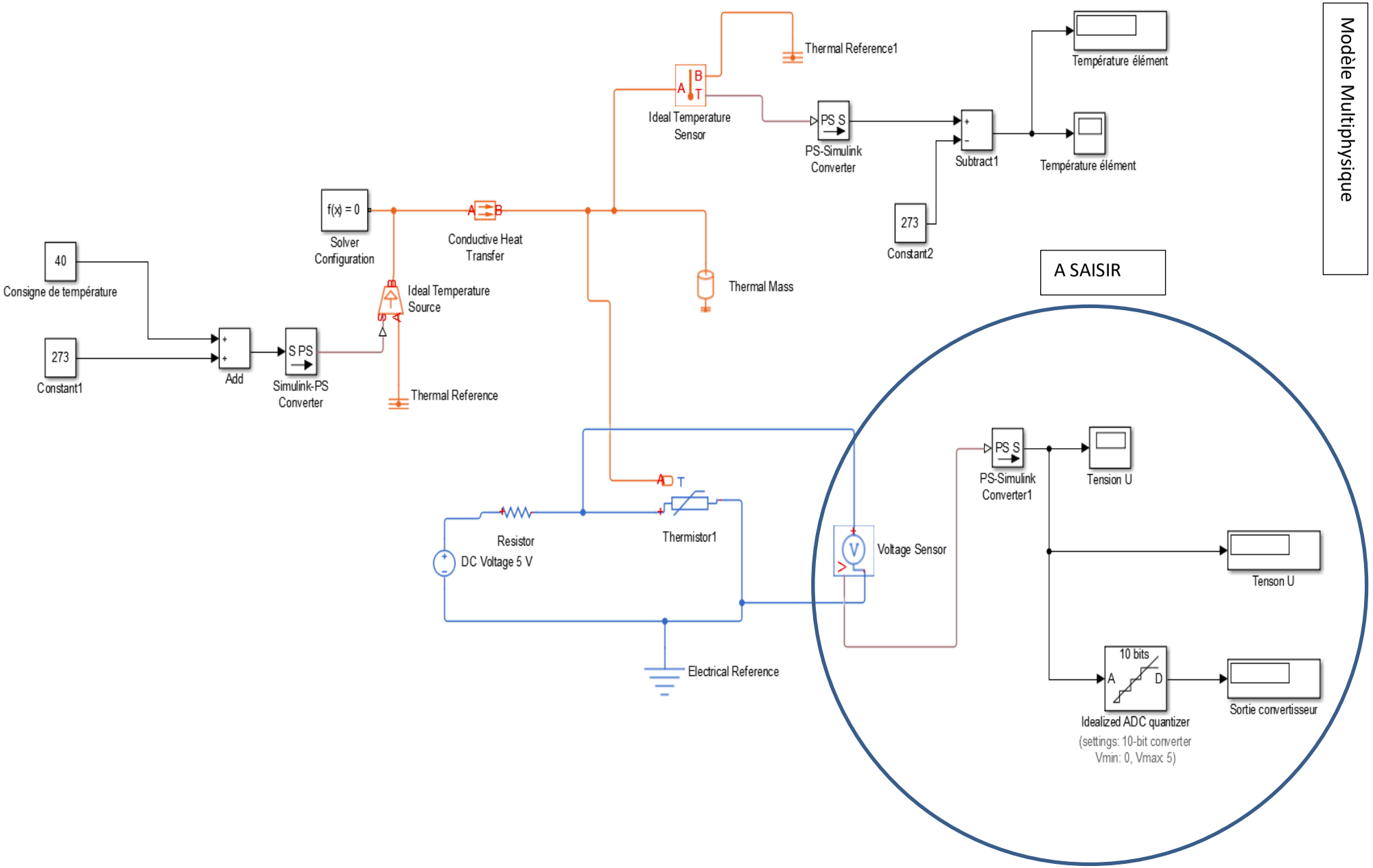
**Charger** le programme puis **tester** le serveur à l'aide d'un autre ordinateur (ouvrir un navigateur puis saisissez l'adresse IP du serveur dans l'URL).

**2.3 Recâbler** le capteur puis modifier le programme afin de transmettre la valeur de la température sur la page html du serveur.

Pour cela intégrer le code vu en 2.3.1 dans le loop du programme. Valider le fonctionnement.

**2.4 Personnaliser** votre page Web en y insérant au moins une image, un titre et un paragraphe en utilisant de la couleur. Vous serez amené à utiliser des liens externes pour la source de l'image, le caractère " devra être « échappé » par un caractère « \ ». Exemple « src=\"https: » pour coder « src="https: »

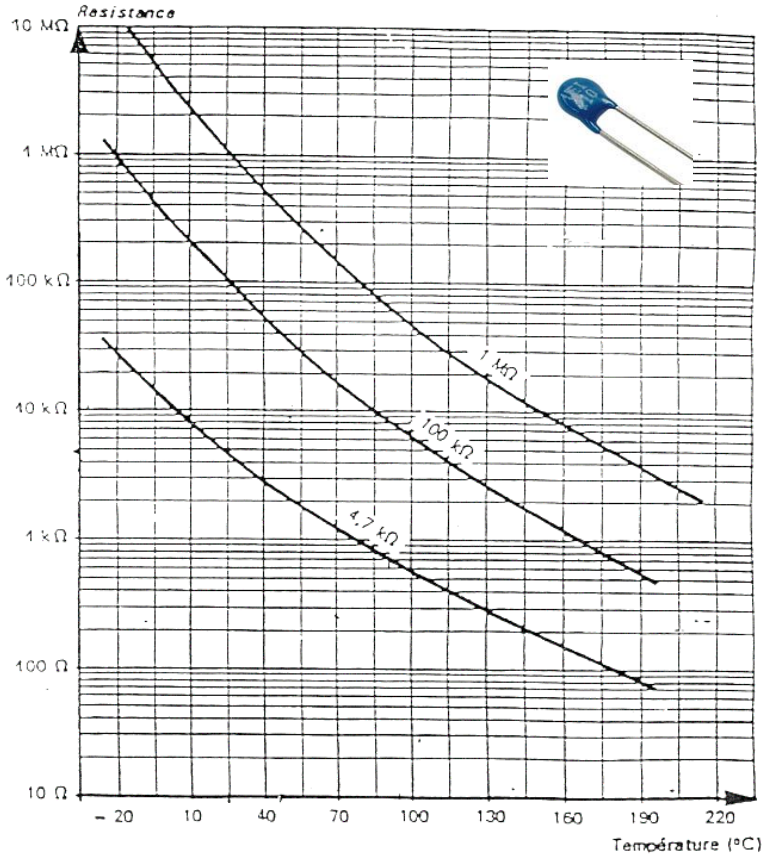
**2.5 Insérer** le code permettant de rafraichir automatiquement la page toute les secondes.


Modèle Multiphysique
A SAISIR
Thermal Reference1
Température élément
Ideal Temperature Sensor
PS-Simulink Converter
Subtract1
Température élément
f(x) = 0
Solver Configuration
Conductive Heat Transfer
273
Constant2
40
Consigne de température
Ideal Temperature Source
Thermal Mass
273
Constant1
Add
Simulink-PS Converter
Thermal Reference
Resistor
DC Voltage 5 V
Thermistor1
Voltage Sensor
PS-Simulink Converter1
Tension U
Tenson U
10 bits
Idealized ADC quantizer
(settings: 10-bit converter
Vmin: 0, Vmax: 5)
Sortie convertisseur
Electrical Reference

***DT1***

***CARACTERISTIQUE DES THERMISTANCES de TYPE CTN***

Les thermistances sont fabriquées à partir d'une combinaison de métaux et de matériaux à base d'oxyde métallique. Elles présentent généralement des coefficients de température négatifs (aussi appelé thermistance CTN), ce qui signifie que leur résistance décroît au fur et à mesure que la température augmente.




***DT2***

***CONVERTISSEUR ANALOGIQUE-NUMERIQUE DE L'ARDUINO***

## Analog-to-Digital Converter

### Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 µs Conversion Time
- Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels (TQFP and QFN/MLF Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - $V_{CC}$ ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

***DT 3***

## analogRead()

Description

Reads the value from the specified analog pin. The Arduino board contains a 6 channel (8 channels on the Mini and Nano, 16 on the Mega), 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit. The input range and resolution can be changed using analogReference().

It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.