



Sciences de l'Ingénieur	Compétences travaillées :	Connaissances associées :
Durée : 2*2 h	- Analyser le traitement de l'information ;	- Algorithme, programme ;
Référence : Hydrao_informations	- Analyser des résultats d'expérimentation et de simulation ;	- Langage informatique ;
Matériel : Système didactique HYDRAO DIDACT ; Carte Arduino Uno ; Capteur CTN 100	- Prévoir l'ordre de grandeur de la mesure ;	- Lois physiques associées au fonctionnement d'un produit ;
Logiciels : Tinkercad pour la simulation ; Arduino IDE ; Edupython	- Identifier les erreurs de mesure ;	- Description qualitative et quantitative des grandeurs physiques caractéristiques du fonctionnement d'un produit ;
Documents : Dossier technique du système ; Dossier ressources	- Conduire des essais en toute sécurité à partir d'un protocole expérimental fourni ;	- Critères de performances ;
	- Instrumenter tout ou partie d'un produit en vue de mesurer les performances ;	- Gamme d'appareils de mesure et capteurs ;
	- Mettre en œuvre une simulation numérique à partir d'un modèle multi-physique pour qualifier et quantifier les performances d'un objet réel ou imaginé	- Règle de raccordement des appareils de mesure et des capteurs ;
	- Rendre compte de résultats ;	- Capteurs, composants d'une chaîne d'acquisition ;
	- Rompre avec l'existant ;	- Paramétrage d'une chaîne d'acquisition ;
	- Améliorer l'existant.	- Carte micro – contrôleur ;
	Pré-requis :	- Paramètres de simulation : durée, incrément temporel, choix des grandeurs affichées, échelles adaptées à l'amplitude et la dynamique des grandeurs simulées ;
	- Premiers éléments sur les capteurs et composants d'une chaîne d'acquisition ;	- Tableau, graphique, diaporama, carte mentale.
	- Premiers éléments sur la programmation en python ;	
Document attendu	Un compte rendu numérique est demandé pour chaque équipe.	

Problème scientifique et technologique :

Le système de douche intelligent imaginé par Hydrao pour réduire le volume d'eau consommé lors d'une douche, et conséquemment l'énergie nécessaire pour chauffer cette eau, repose à la fois sur une limitation du débit d'eau, et l'implication de l'utilisateur en l'informant de ses consommations.

On se propose dans cette activité d'étudier la chaîne d'information du produit et notamment la captation et la mesure du volume d'eau consommée, puis nous envisagerons une évolution du produit qui permettrait par la mesure de la température d'estimer précisément l'énergie de chauffage de l'eau et de prévenir le risque de brûlure.

1ère partie : l'architecture fonctionnelle

2ème partie : conditionnement des signaux et mesure du volume écoulé

3ère partie : Mesurer une température avec le capteur TMP36.

4ème partie : Mesurer une température avec le capteur CTN 100.

5ème partie : Intérêt de mesurer la température



Présentation générale du système

Le système de douche intelligent d'Hydrao est composé d'un pommeau de douche (« Aloé » dans notre cas) autoalimenté, et connecté à un smartphone où l'application « HYDRAO SMART SHOWER » est installée : <https://www.hydrao.com/fr/>.



Principe de fonctionnement :

- La couleur des LED éclairant le jet d'eau change avec le volume d'eau écoulé, informant l'utilisateur de sa consommation d'eau en temps réel ;
- Grâce à une application installée dans un smartphone, l'utilisateur autorisé peut consulter l'historique des consommations, personnaliser et paramétrer certains éléments du système ;
- Les données anonymisées sont transmises à un cloud, permettant des analyses statistiques ;
- Une turbine-génératrice assure l'auto-alimentation électrique du pommeau ; le pommeau ne comporte ni pile ni batterie.

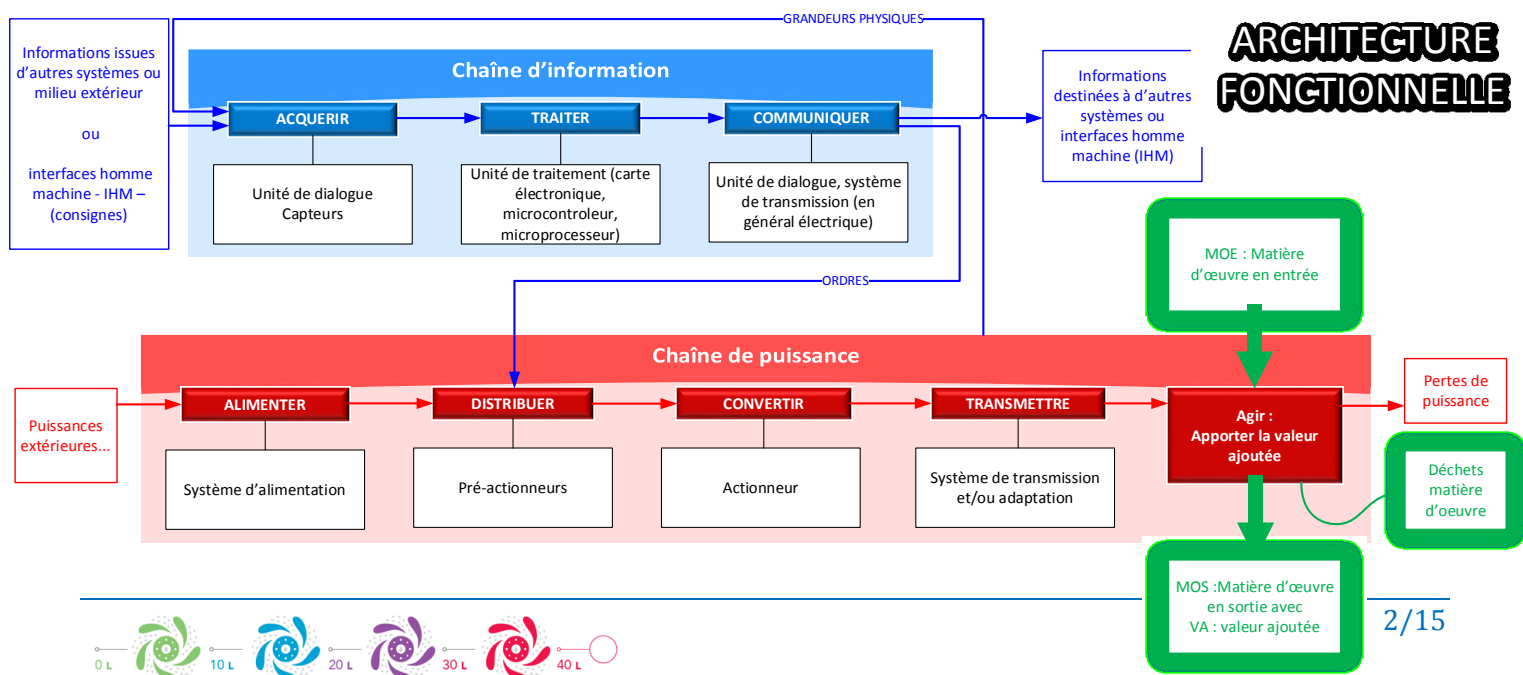
Problème technique :

Comment le pommeau Hydrao mesure la quantité d'eau écoulée ?

Quelles informations sont captées, comment sont-elles traitées ?

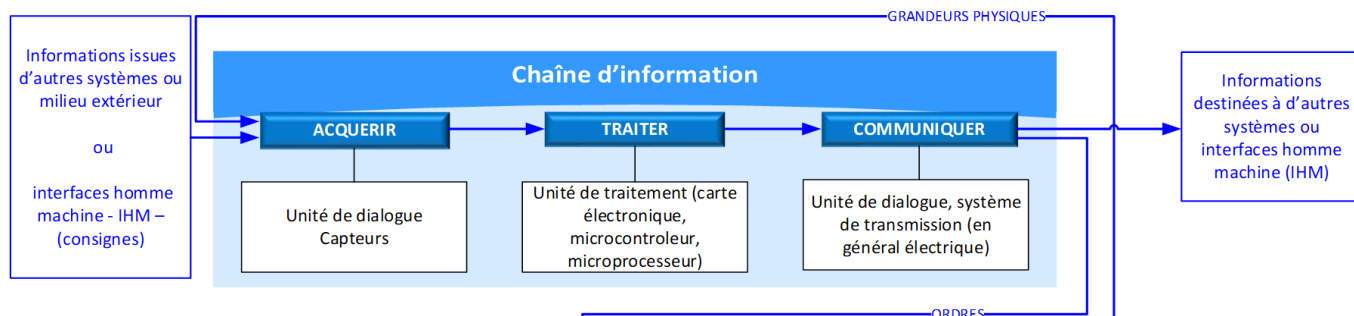
1ère partie : l'architecture fonctionnelle

On adopte ici à une approche fonctionnelle. L'outil d'architecture fonctionnelle permet la description d'un système pluri-technologique :





La chaîne de puissance qui permet l'alimentation électrique du pommeau n'est pas abordée ici, on s'intéresse donc à la partie chaîne d'information :



Activité : observez le fonctionnement du pommeau de douche en situation, si possible pommeau et smartphone avec application hydrazo.

Quelles sont les informations acquises du milieu extérieur ?

Quelles sont les informations renvoyées à l'extérieur (informations destinées à d'autres systèmes ou interfaces homme machine (IHM)) ?

Remplissez les cadres correspondants de la figure 1 :

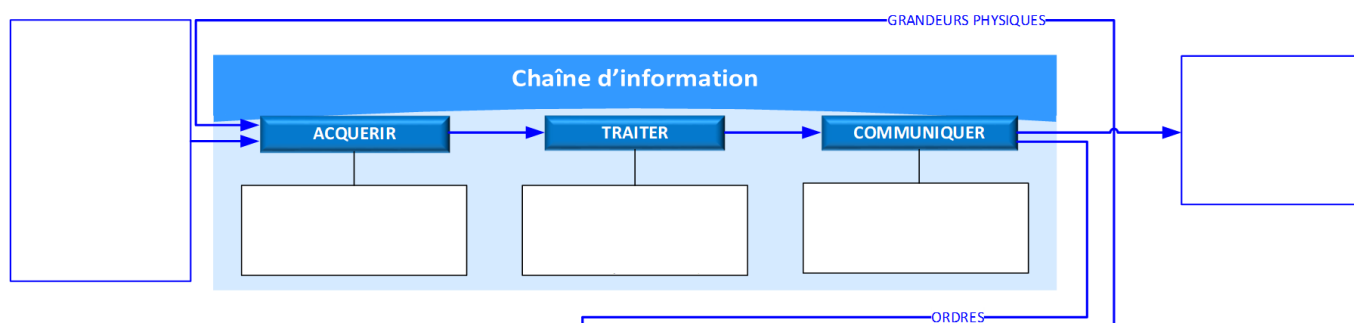


Figure 1 : chaîne d'information à compléter

Faites une recherche :

Quel type de composant permet de mesurer un débit ? Comment ce composant fonctionne ?

Ouvrez le document ressource « Hydrazo_sysml.pdf »

Recherchez dans les diagrammes SysML le composant qui permet de mesurer le débit d'eau consommée, celui ou ceux qui traitent les signaux électriques issus de ce composant, et celui ou ceux qui permettent la communication vers la chaîne de puissance. Finissez de compléter la figure 1.

2ème partie : conditionnement des signaux et mesure du volume écoulé

A- Travail préparatoire

Observez plus précisément le diagramme SysML idd (blocs internes). On remarque que 2 grandeurs sont prélevées : Upuiss et Ugene. Dans l'étude énergétique du produit, on a constaté que :

- Ugene qui est la tension mesurée par la génératrice, est un signal périodique sinusoïdal,
- Upuiss qui est la tension de la génératrice redressée et filtrée, est un courant continu.

Regardez ensuite le schéma synoptique du document « Hydrazo_sysml.pdf », repérez Ugene et Upuiss et expliquez en quelques phrases quelles informations peuvent apporter ces deux grandeurs pour la mesure du volume écoulé.



B- Expérimentations sur la carte Hydrao.

Vous allez avoir besoin :

- Des cartes de prototypage Hydrao puissance et Hydrao LED.
- D'un générateur de signal (GBF).
- D'un transformateur de découplage.
- D'un oscilloscope muni de ses sondes (pointes de touches ou de mini-accroche-fil).
- de fils de prototypage.
- 4 cordons « banane » et 2 pinces.

Enfichez la carte LED sur la carte de puissance :

Branchez 2 fils sur le bornier de la carte de puissance.

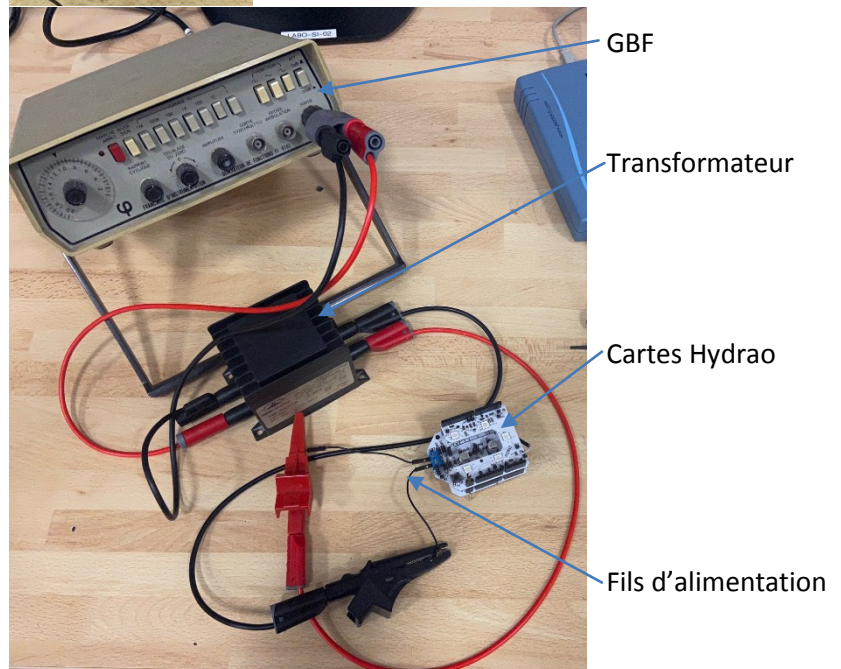
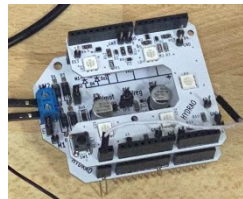
Assurez-vous que le GBF est éteint.

Branchez le transformateur sur la sortie du GBF, puis la sortie du transformateur aux fils d'alimentation de la carte grâce aux pinces.

Réglez le GBF de façon à ce qu'il produise un signal sinusoïdal de fréquence environ 10 Hz, amplitude environ 3V à 5V.

Ce signal simule le fonctionnement du générateur lorsque la turbine est en action. Préparez l'oscilloscope et ses sondes pour la mesure :

Faites vérifier le montage par le professeur.



Allumez le GBF.

Affichez sur l'oscilloscope et caractérisez les signaux suivants (pensez à faire des copies d'écran pour vos comptes-rendus) :

- Tension d'entrée (sur le bornier l'alimentation)
- Broche Vmot
- Broche A3 (AD3)
- Broche M1
- Broche A0

Une fois les mesures faites, éteignez le GBF.



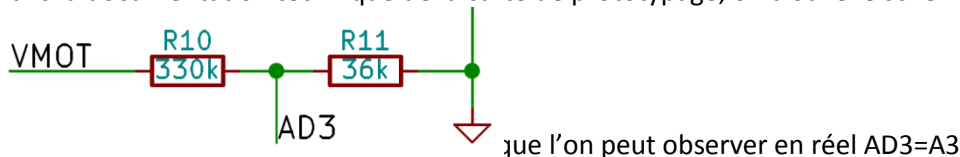
Regardez de nouveau le schéma synoptique page 9 du document SysML.

Parmi vos mesures, repérez Ugene, Upuiss, image_Upuiss et creneaux_gene.



C- Présence de Upuiss

Dans la documentation technique de la carte de prototypage, on trouve le schéma suivant :



Quel montage classique reconnaissez-vous ?

Donnez l'expression de la tension en AD3 en fonction de R10, R11 et VMOT.

Calculez le coefficient multiplicateur permettant de passer de VMOT à la tension en AD3.

AD3 (=A3) est un pin d'entrée analogique (lecture) du microcontrôleur. Dans le produit d'origine, le microcontrôleur est un STM32 de chez ST Microelectronics qui est équipé d'un convertisseur analogique-numérique (CAN) qui accepte des tensions allant jusqu'à 3,3 V.

Expliquez l'utilité du montage ci-dessus. A quel bloc du schéma synoptique correspond ce montage ?

La valeur du coefficient multiplicateur calculé précédemment vous semble-t-elle correcte ?

Quelle est l'information apportée par la présence ou non d'un courant en AD3 ?

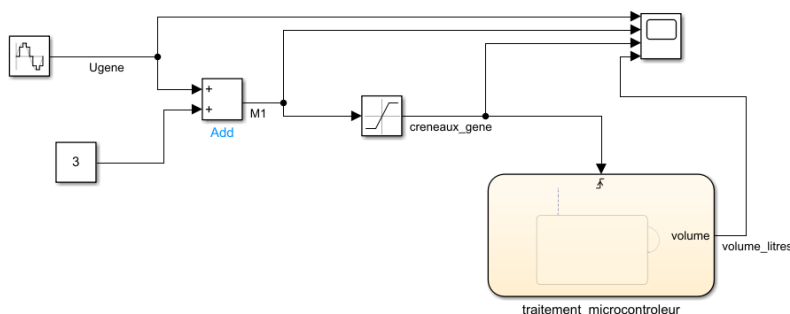
D- Traitement de Ugene

Copiez le fichier

Hydrao_traitement_signal_gene_XXXX.slx dans Documents/MATLAB

Ouvrez-le à partir de cet emplacement.

Exécutez cette simulation Simulink, si le scope ne s'ouvre pas ouvrez-le et observez les signaux. Comparez-les avec les signaux mesurés à l'oscilloscope en début d'activité.



On remarque que l'entrée du diagramme d'état est un événement de type front montant.

A quoi correspond sur le pommeau, la durée qui sépare 2 événements consécutifs sur ce port.

Ouvrez le diagramme d'états « traitement_microcontrôleur ».

Expliquez comment à partir du signal sinusoïdal le pommeau hydrao évalue la quantité d'eau consommée. La formule du traitement informatique du diagramme d'états est-elle cohérente avec les données présentes dans les diagrammes SysML, justifiez.

E- Performance de la mesure

Vous allez avoir besoin :

- du pommeau Hydrao et son flexible branché sur un robinet,
- d'un smartphone avec bluetooth et l'application « SMART AND BLUE » Hydrao installée.
- d'un récipient de type seau gradué à 5 litres ou 10 litres.





Branchez le pommeau Hydrao sur un robinet puis ouvrez l'application Hydrao et connectez le pommeau en bluetooth au téléphone.

Remplissez le seau plusieurs fois avec le pommeau pour imiter une douche. La pause entre deux remplissages ne doit pas dépasser 2 min.

Poursuivez l'opération jusqu'à atteindre 50 litres consommés.

A chaque seau, notez l'état du pommeau (couleur d'éclairage) ainsi que le volume affiché.

Attendre plus que 2 min pour que la consommation se réinitialise puis recommencer l'essai.

Le comportement observé (changement de couleur) est-t-il identique à celui de la machine d'état technologique du SysML ?

Pour chaque essai, calculer l'écart entre le volume affiché et le volume réel (=somme des volumes des seaux).

La mesure de volume écoulé par le pommeau est-elle juste ?

Analyser les valeurs de volume affichées pour chaque seau ou pour des séries d'essais différentes.

La mesure de volume écoulé par le pommeau vous semble-t-elle fidèle ?



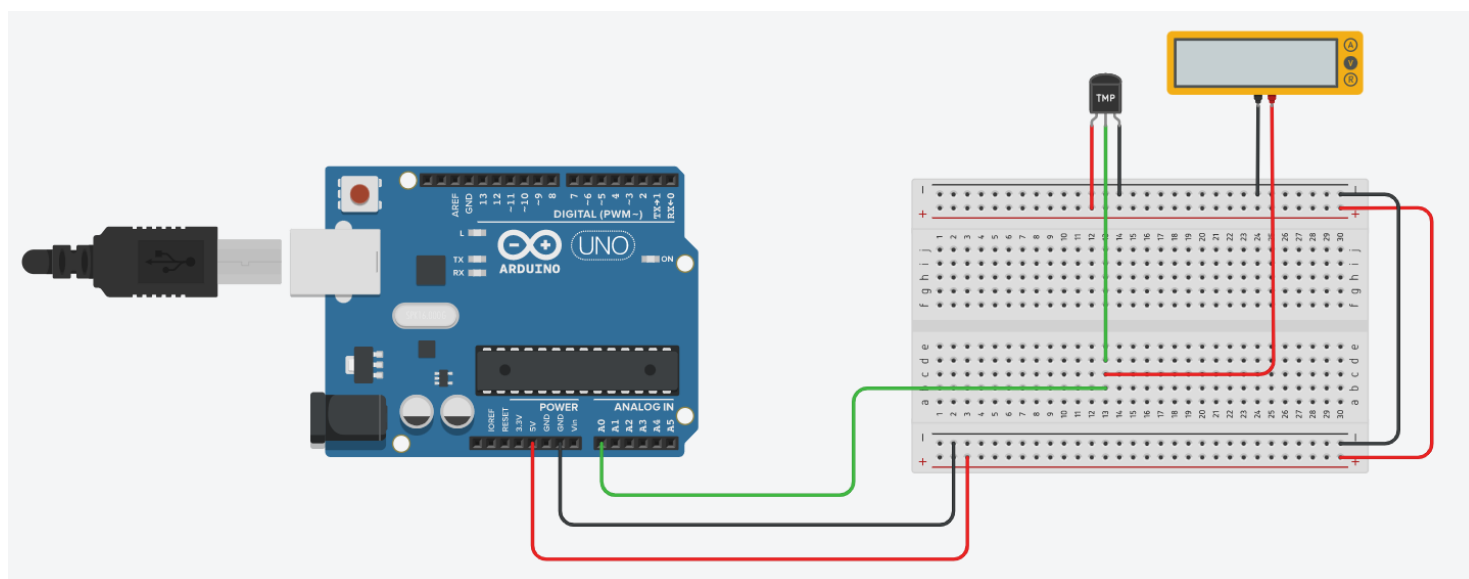
Problème technique :

Nous allons étudier maintenant une évolution du produit qui permettrait par la mesure de la température d'estimer précisément l'énergie de chauffage de l'eau et de prévenir le risque de brûlure.

3ème partie : mesure d'une température avec le capteur TMP36

Le TMP36 est un capteur de température dont la tension de sortie est proportionnelle à la température mesurée.

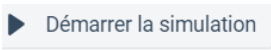
- 1- A l'aide des données techniques (datasheet) du **document ressource 1** (DR1), énoncer la plage de température mesurée par ce capteur.
- 2- A l'aide du **document ressource 2**, se connecter sur le logiciel en ligne « Tinkercad » pour créer un circuit électrique à l'aide des bibliothèques disponibles, de le programmer en blocs et de réaliser sa simulation.
- 3- Réaliser le montage ci-dessous sur « Tinkercad » en utilisant les composants suivants :
 - Une carte programmable : Arduino Uno ;
 - Un capteur de température : TMP36 ;
 - Un multimètre configuré en voltmètre.

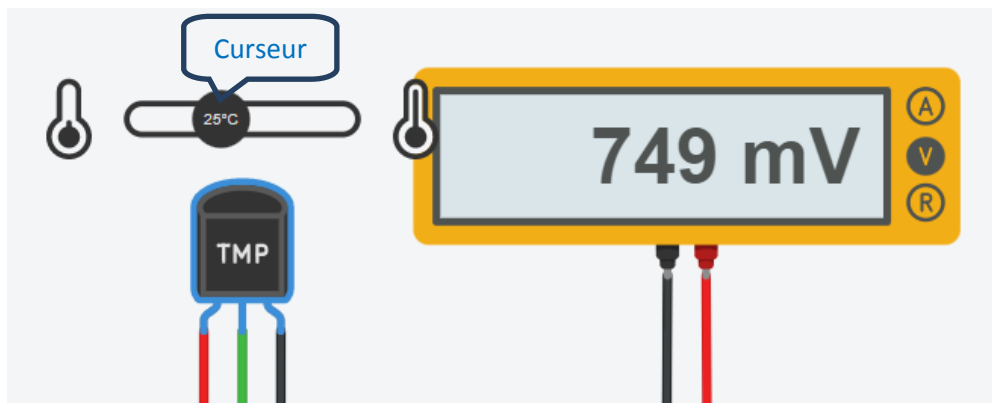


Remarques :

- Le capteur TMP 36 possède 3 fils : Alimentation (+5v), tension de sortie et la masse (-).
 - Le capteur TMP36 est branché à l'entrée analogique A0 du microcontrôleur qui ne mesure pas directement la valeur de la tension aux bornes du capteur. Le microcontrôleur possède un convertisseur analogique numérique sur 10 bits. Au niveau de l'entrée analogique A0, cette tension est convertie en un nombre qui peut prendre n'importe quelle valeur entière entre 0 et 1023 (points). La valeur de 1023 est associée à une valeur maximale de 5V.
 - Un voltmètre est donc nécessaire pour mesurer la tension aux bornes du capteur.
- 4- Tracer l'évolution de la température en degrés Celsius en fonction de la tension en volts mesurée aux bornes du capteur TMP36.

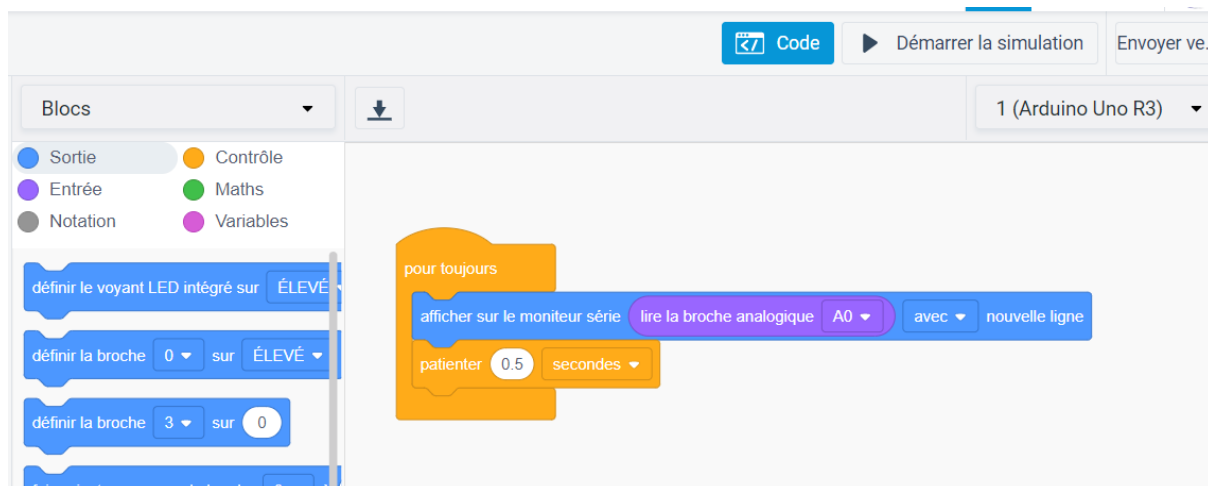


- Démarrer la simulation : 
- Zoomer votre montage et sélectionner le capteur TMP36, un curseur apparaît :



- Relever les couples de mesures **Température (°C) / Tension (V)** pour la plage -40°C à +125°C, de 10°C en 10°C que vous noterez dans le tableau « Temperature_Tension_A0.ods » situé dans votre dossier ressource ;
- La courbe de la **Température (°C) en fonction de la Tension (V)** est apparue sur le tableur, la reporter.

- 5- A l'aide de la courbe ci-dessus, quelle est la plage de tension utile du capteur TMP36 ?
- 6- Connaissant les plages de température et de tension du capteur TMP36 (voir courbe ci-dessus), en déduire sa sensibilité (variation du signal de sortie par rapport à la variation du signal d'entrée en mV/°C). Vérifier avec la valeur de la datasheet.
- 7- A l'aide de la courbe ci-dessus, relever la valeur de la tension à 25°C, la comparer à la datasheet.
- 8- A l'aide de l'équation de la courbe ci-dessus, retrouver la tension d'offset = 0,5V de la datasheet.
- 9- Réaliser le programme en blocs ci-dessous :
 - Sélectionner « Code » ;
 - Glisser-déposer les blocs choisis.



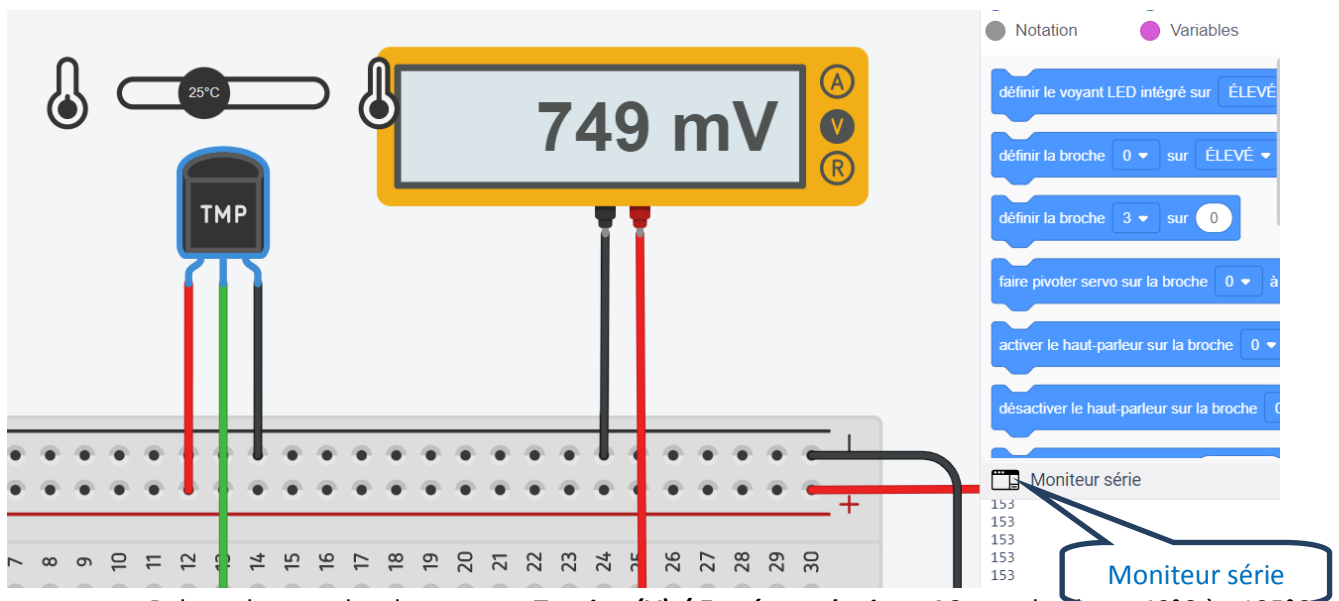


Remarque :

- Ce programme va se répéter indéfiniment, il va envoyer sur le moniteur série toutes les 0,5 s la valeur reçue sur l'entrée analogique A0.

10- Tracer l'évolution de la tension en volts en fonction de l'entrée analogique A0, mesurée aux bornes du capteur TMP36 et convertie par le microcontrôleur :

- Démarrer la simulation :
- Sélectionner : Code
- Sélectionner : Moniteur série



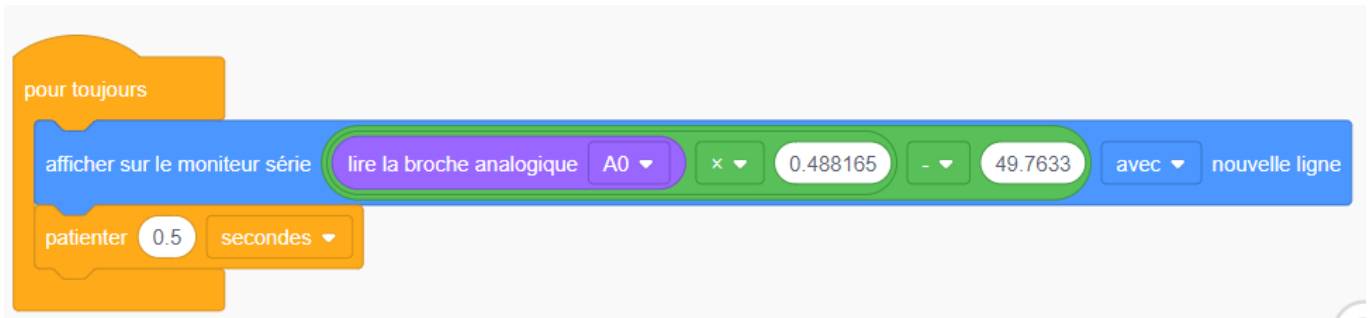
- Relever les couples de mesures **Tension (V) / Entrée analogique A0** pour la plage -40°C à $+125^{\circ}\text{C}$, de 10°C en 10°C que vous noterez dans le tableau « Temperature_Tension_A0.ods » situé dans votre dossier ressource ;
- La courbe de la **Tension (V) en fonction de A0** est apparue sur le tableur, la reporter.
 - A l'aide de la courbe ci-dessus, quelle est la plage de l'entrée analogique A0 ?
 - A quoi correspond le coefficient directeur de la courbe ci-dessus ?
 - Quelle est la plus petite valeur de tension mesurée par ce capteur ?
 - Connaissant les plages de l'entrée analogique A0 et de température du capteur TMP36 (voir courbe ci-dessus), quelle est la résolution de ce capteur (plus petite valeur de température mesurable) ?
 - En déduire la précision de ce capteur et la comparer à celle de la datasheet (voir DR1) :

11- A l'aide des courbes précédentes, donner l'équation de la température ($^{\circ}\text{C}$) en fonction de l'entrée analogique A0.

12- La courbe de la **Température (°C) en fonction de A0** est apparue sur le tableur, la reporter.

13- Réaliser le programme en blocs ci-dessous :

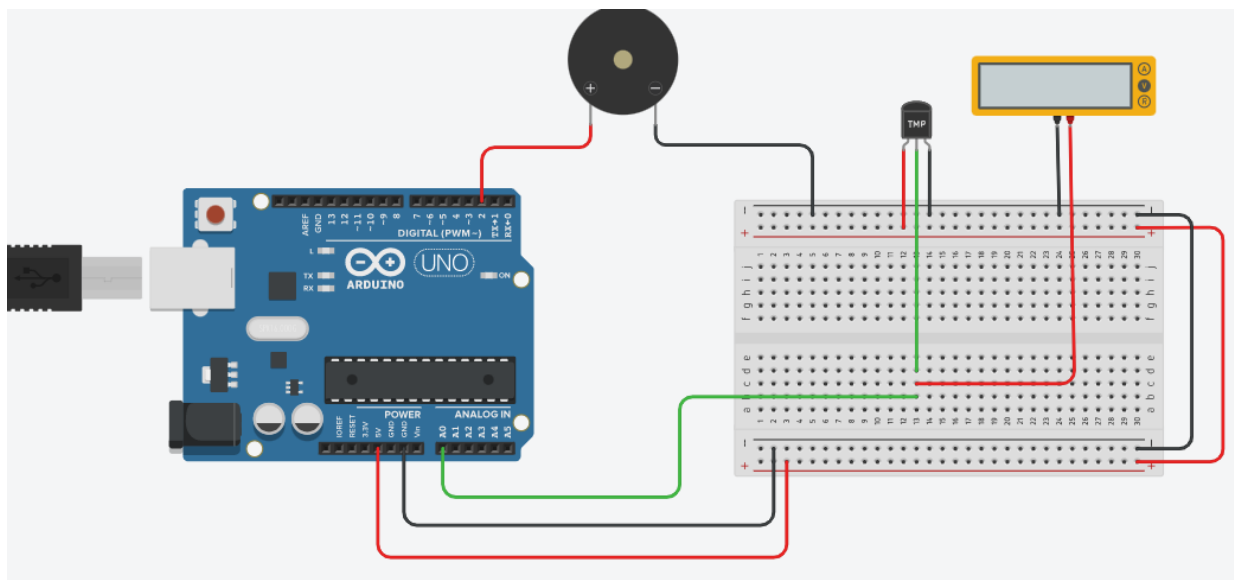
- Sélectionner « Code » ;
- Glisser-déposer les blocs choisis :



- Expliquer ce que va faire ce programme :
- Démarrer la simulation et tester ce nouveau programme.

14- Si l'on considère que la température idéale pour une douche doit être légèrement supérieure à celle de notre corps puisqu'elle est située entre 40 et 43 degrés, modifier le programme précédent pour être informé par un buzzer en cas de dépassement de la température que l'on se fixe à 43 °C :

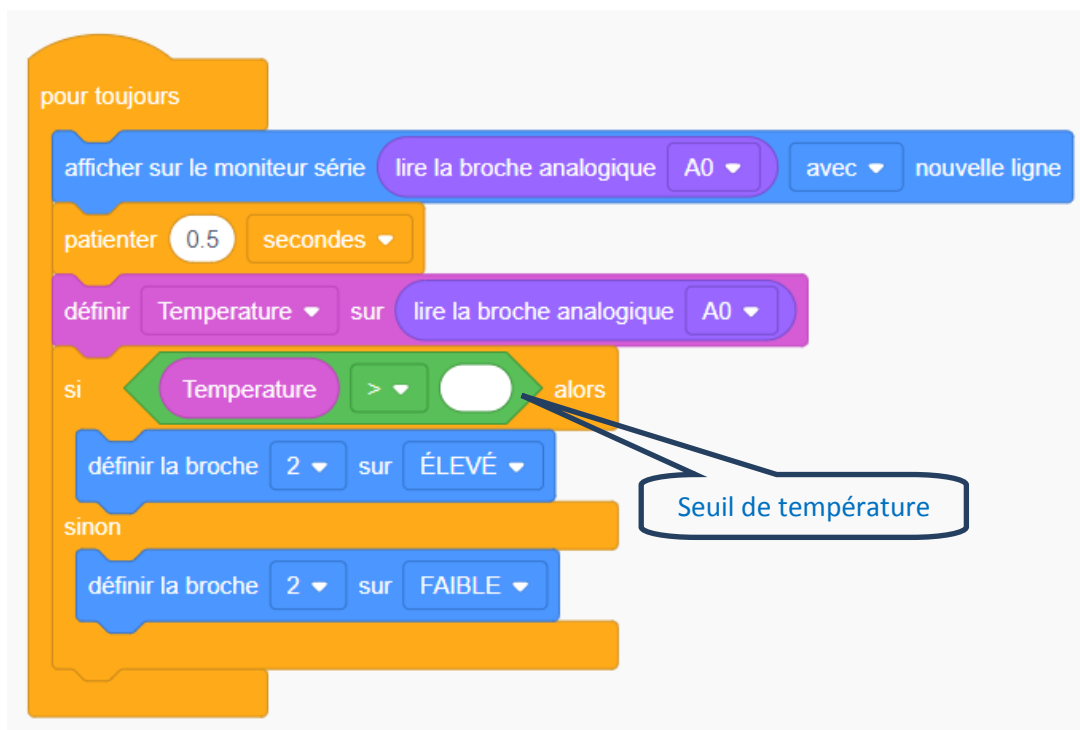
- Trouver la valeur analogique A0 correspondant à une température de 43°C :
- Modifier votre montage en rajoutant les composants suivants :
 - ✓ Un buzzer connecté sur la sortie digital D2;
 - ✓ 2 fils.



- Modifier le code comme ci-dessous :
 - ✓ Créer une variable « Temperature » ;



- ✓ Activer le buzzer si la « Temperature » est supérieure au seuil défini précédemment ;
- ✓ Sinon Désactiver le buzzer.

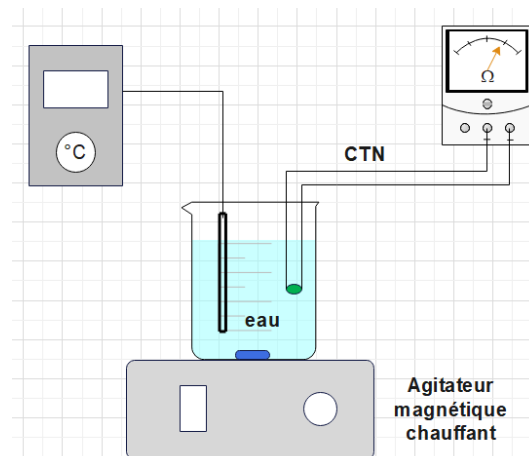


4ème partie : Mesure d'une température avec une CTN

Une CTN est une thermistance, c'est-à-dire un composant électronique dont la résistance électrique varie en fonction de la température.

15- Réaliser le montage ci-contre :

- Lister les appareils de mesures ainsi que les grandeurs mesurées :



16- Etalonner le capteur en suivant le protocole expérimental ci-dessous :

- Ajouter quelques glaçons dans l'eau et attendre quelques minutes que la température initiale diminue (environ 5°C) ;
- Allumer l'agitateur magnétique chauffant ;
- Relever les couples de mesures **Température (°C) / Résistance CTN (Ω)** pour la plage 5°C à 90°C, de 5°C en 5°C que vous noterez dans le tableau « Temperature_Resistance.ods » situé dans votre dossier ressource ;



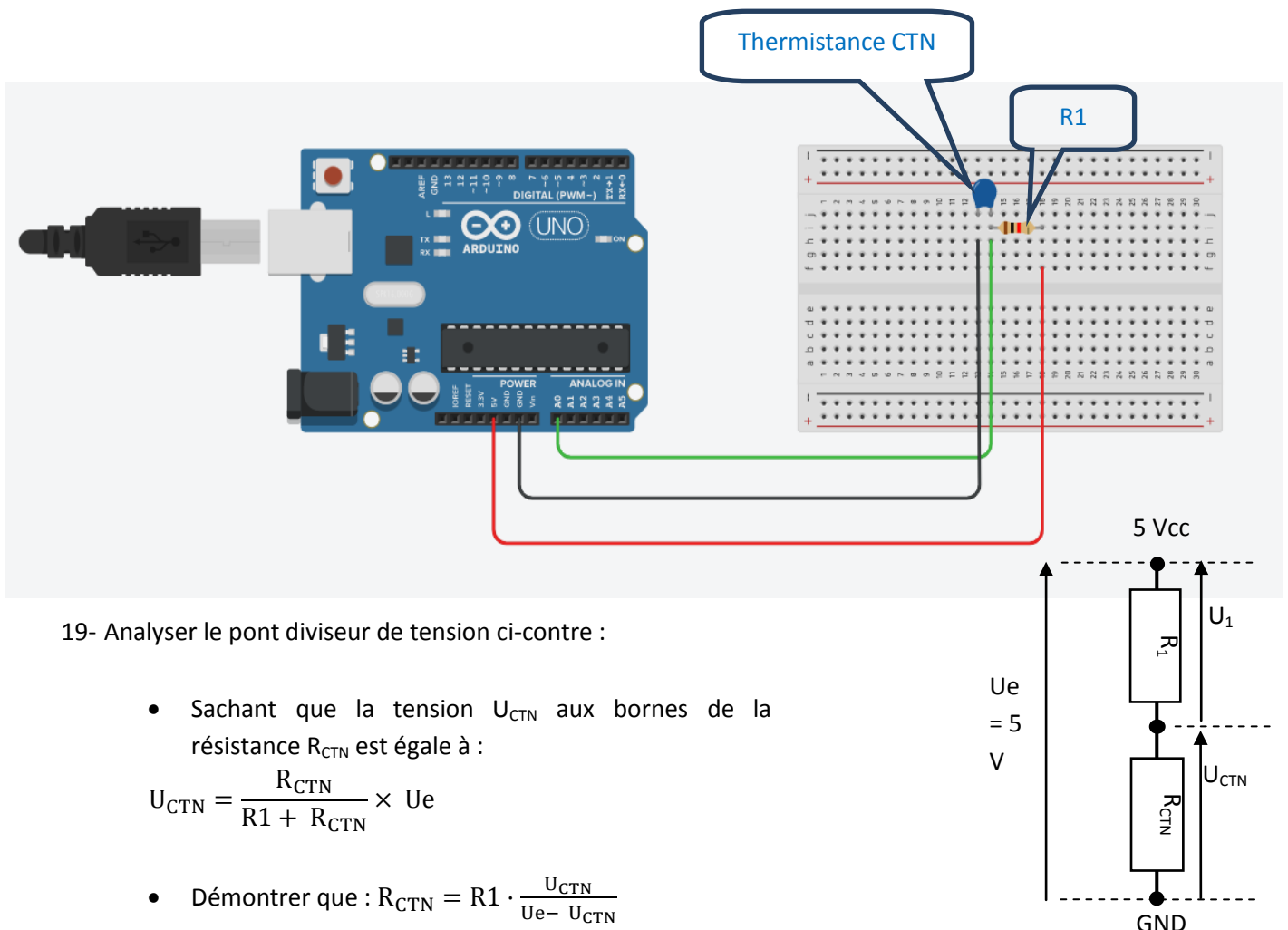
- La courbe d'étalonnage de la **Température (°C) en fonction de la Résistance CTN (Ω)** est apparue sur le tableur, la reporter : $T = f(R_{CTN})$.
- Déterminer l'équation de la courbe obtenue, quelle est la loi de cette courbe ?

17- Récupérer les composants suivants :

- Une carte programmable : Arduino Uno ;
- Une CTN100 ;
- Une résistance R1 = 1 kΩ ;
- 3 fils.

18- Réaliser le montage ci-dessous en suivant le protocole suivant :

- Brancher le capteur CTN sur l'entrée analogique A0 du microcontrôleur. Le microcontrôleur ne mesure pas directement la valeur de la tension U_{CTN} en volts, il renvoie un nombre décimal compris entre 0 et 1 qui correspond à une tension comprise entre 0V et 5V pour la tension U_{CTN} .
- Associer la CTN en série avec une résistance R1 égale à 1 kΩ. Une tension U_e de 5 V est appliquée aux bornes de cette association. Le montage obtenu est appelé « un pont diviseur de tension ».



19- Analyser le pont diviseur de tension ci-contre :

- Sachant que la tension U_{CTN} aux bornes de la résistance R_{CTN} est égale à :

$$U_{CTN} = \frac{R_{CTN}}{R_1 + R_{CTN}} \times U_e$$

- Démontrer que : $R_{CTN} = R_1 \cdot \frac{U_{CTN}}{U_e - U_{CTN}}$



20- Réaliser le programme en langage python en installant le firmware « Firmata -> StandardFirmata » sur la carte Arduino, puis Télécharger la bibliothèque « Pyfirmata » dans Edupython (voir dossier ressource DR3) :

- Ci-dessous le programme à saisir sur Edupython permettant de mesurer la température :

```

1  from pyfirmata import Arduino, util
2  import time
3  import math
4  carte = Arduino('COM9')
5  acquisition = util.Iterator(carte)
6  acquisition.start()
7  val_A0 = carte.get_pin('a:0:i')
8  time.sleep(1.0)
9  R1 = 1000 #Ohm
10 for i in range(0,60):
11     tensionCTN = 5*val_A0.read()
12     Rctn =
13     temperature =
14     print(temperature, "°C")
15     time.sleep(1)
16 carte.exit()
```

- Quelle ligne du programme permet la lecture de la variable **val_A0** sur l'entrée A0 du microcontrôleur ?
- Quelle ligne du programme permet la lecture de la variable **tensionCTN** ?
- Compléter la ligne 12 du programme permettant au programme de calculer la valeur de la résistance et la stocker dans la variable **Rctn** (voir Q.19) ;
- Compléter la ligne 13 du programme permettant au programme de calculer la valeur de la **température** à partir de la résistance **Rctn** (voir Q.16) ;
- Quelle ligne du programme permet d'envoyer la valeur de la température calculée sur la console python :
- Exécuter le programme. Attention, il peut être nécessaire de modifier la ligne 4 en précisant le port sur lequel le microcontrôleur est connecté à l'ordinateur ;
- Prendre la CTN dans la main et observer l'évolution de la valeur de la température indiquée dans la console python.



21- Afficher un graphique en temps réel de la température (°C) en fonction du temps avec Matplotlib :

```

1  from pyfirmata import Arduino, util
2  import matplotlib.pyplot as plt
3  import time
4  import math
5  carte = Arduino('COM9')
6  acquisition = util.Iterator(carte)
7  acquisition.start()
8  val_A0 = carte.get_pin('a:0:i')
9  time.sleep(1.0)
10 T, t = [],[]
11 durée_mesure = 120 #en secondes
12 interval = 2 #en seconde
13 nb_mesure = int(durée_mesure/interval)
14 R1 = 1000
15 ti=time.time()
16 print("Début du test")
17 for i in range(0,nb_mesure):
18     plt.cla()
19     tensionCTN = 5*val_A0.read()
20     Rctn = R1*tensionCTN/(5-tensionCTN)
21     temperature = 169.1-29.82*math.log(Rctn)
22     T.append(temperature)
23     t.append(time.time()-ti)
24     plt.axis([0,durée_mesure,0,100])
25     plt.plot(t,T, "r+")
26     plt.pause(0.01)
27     time.sleep(interval)
28 plt.show()
29 print("Fin du test")

```

Matplotlib est une bibliothèque de Python permettant d'afficher des graphiques, nous allons l'utiliser pour afficher les mesures effectuées par la carte en temps réel.

Initialisation de la carte

Création des 2 listes pour les 2 séries de mesures (Température et temps)

Déclaration de la durée de la série de mesure

Déclaration de l'intervalle de mesure

Les mesures de temps se feront avec la fonction `time.time()`, qui renvoie un nombre de seconde par rapport à une référence de l'ordinateur, il faut donc mesurer un temps initial et le

Mesure de la tension CTN par la carte et calcul de la température

Ajout des mesures dans les

Définition des axes du graphique

Affichage du graphique en temps réel, « r+ » pour des croix

Temps d'attente entre chaque mesure

Tracé de la courbe



5ème partie : Intérêt de mesurer la température d'une douche

22- Le pommeau hydrao est une solution ludique pour encourager l'utilisateur à diminuer sa consommation d'eau, en quoi la mesure de la température de l'eau peut participer à l'effort d'économie ?

Données : 0,15€/KWh ; capacité calorifique de l'eau : $C_p = 4,185 \text{ J} \cdot \text{Kg}^{-1} \cdot \text{K}^{-1}$; $E = m \cdot C_p \cdot \Delta T$

23- Sachant que l'eau est à une température moyenne de 15° C avant d'être chauffée, calculer l'énergie puis le coût d'une douche de 60 litres à 43°C :

24- Grâce au produit Hydrao, l'utilisateur va réduire sa consommation d'eau à 20 litres et à une température de 37 °C. Calculer l'énergie puis le coût d'une douche et en déduire l'économie réalisée sur une année en considérant une douche par jour.

25- En quoi l'évolution du produit hydrao pourrait intégrer une sécurité pour la toilette des nourrissons ?

Sources – liens :

- Le site officiel Arduino : <https://www.arduino.cc/>
- La documentation officielle pyfirmata : <https://pyfirmata.readthedocs.io/en/latest/index.html>
- La physique avec Arduino : <https://opentp.fr/card/>