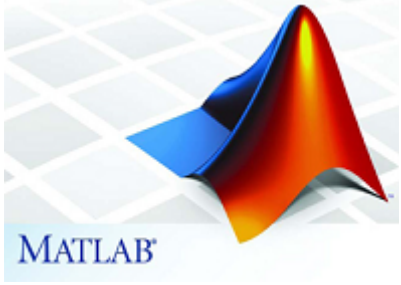
Modélisation avec Matlab

Table des matières



1. Introduction

L'objectif de cette activité est d'expérimenter le chauffage d'un volume de liquide à l'intérieur d'un récipient, de maîtriser l'évolution de la température (rapidité de chauffe et précision) et d'estimer les pertes vers l'environnement extérieur. Le dispositif sera constitué d'un bocal de verre d'un volume de 1l rempli d'eau ; d'un thermoplongeur de 300 W ; d'un relais statique ; d'une carte Arduino et d'une thermistance. La programmation de la carte Arduino sera assurée par le logiciel Matlab et son module Simulink. Par ailleurs, une modélisation du dispositif permettra de vérifier les performances par simulation.

2. Présentation des constituants

2.1. Bocal de verre et thermoplongeur

Conductivité thermique du verre :1 W/(m K).

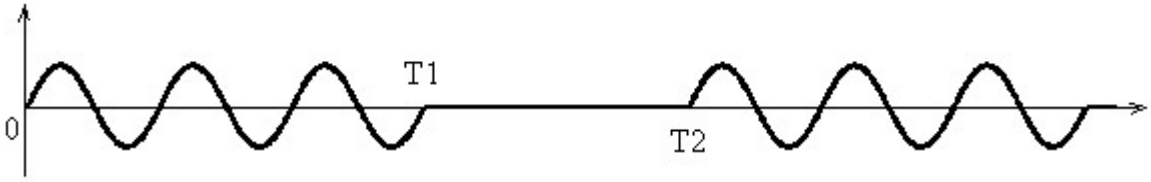
Épaisseur :

Puissance du thermoplongeur : 300W.

Chaleur spécifique de l'eau : 4180 J/Kg/K.

2.2. Relais statique

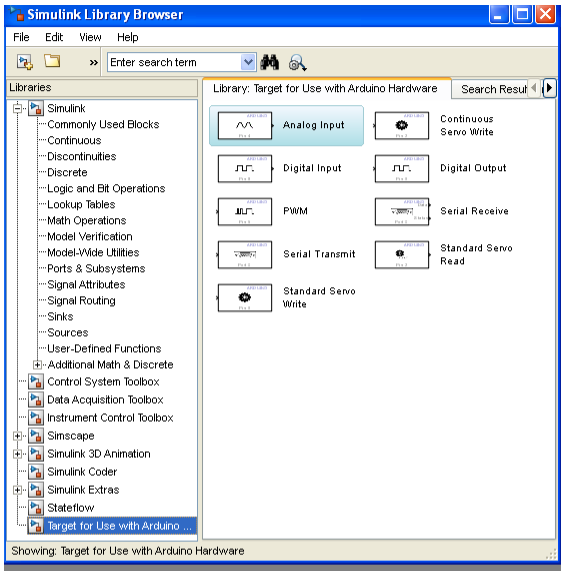
Le bloc statique permet de transmettre l'énergie électrique depuis le réseau vers le thermoplongeur. Il dispose d'une partie puissance qui autorise la commutation de tensions jusqu'à 400V et d'une partie commande qui reçoit les données de la carte Arduino. Un bloc statique permet une modulation de puissance électrique de type PWM . Il va s'agir d'autoriser le passage de la puissance électrique pendant une certaine durée sur une période fixe. Ici nous choisirons une période fixe de 10s. La puissance transmise sera directement proportionnelle au rapport cyclique de la fonction ; c'est à dire au rapport de l'état passant sur la période totale.



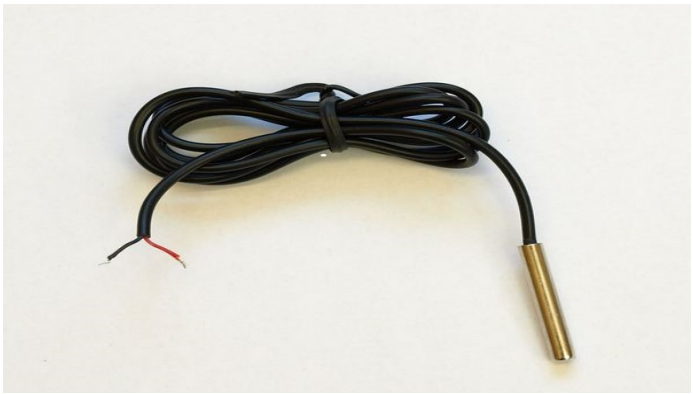
La modulation en trains d'ondes n'est évidemment possible que sur les processus à constantes de temps très supérieures à la période de modulation. C'est bien le cas pour le chauffage.

2.3. La carte Arduino et Matlab

La carte Arduino méga sera programmée par le Simulink, un module de Matlab. La carte méga comme cible de Simulink permet un fonctionnement connecté au PC et facilite l'acquisition des données en cours d'expérience.



2.4. La thermistance CTN



Les CTN (Coefficient de Température Négatif, en anglais. NTC, *Negative Temperature Coefficient*) sont des thermistances dont la résistance diminue de façon uniforme quand la température augmente.

On peut exprimer une relation entre la résistance de la CTN et sa température par la relation de Steinhart-Hart :

$$\frac{1}{T} = A + B\ln(R_T) + C(\ln(R_T))^3$$

Pour trouver les coefficients de Steinhart-Hart il suffit de connaître trois points de fonctionnement et de poser un système. Pour cela, on utilise trois valeurs de résistance données pour trois températures connues.

$$\begin{cases} A + (\ln R_1).B + (\ln R_1)^3.C = \frac{1}{T_1} \\ A + (\ln R_2).B + (\ln R_2)^3.C = \frac{1}{T_2} \\ A + (\ln R_3).B + (\ln R_3)^3.C = \frac{1}{T_3} \end{cases}$$

Avec R1, R2 et R3 les valeurs de la résistance aux températures T1, T2 et T3, on peut alors exprimer A, B et C après quelques substitutions: posons d'abord :

$$L_1 = \ln(R_1) \qquad L_2 = \ln(R_2) \qquad L_3 = \ln(R_3)$$

et,

$$Y_1 = \frac{1}{T_1} \qquad Y_2 = \frac{1}{T_2} \qquad Y_3 = \frac{1}{T_3}$$

puis,

$$\gamma_2 = \frac{Y_2 - Y_1}{L_2 - L_1} \qquad \gamma_3 = \frac{Y_3 - Y_1}{L_3 - L_1}$$

il vient :

$$\Rightarrow C = \left(\frac{\gamma_3 - \gamma_2}{L_3 - L_2}\right) \times \left(\frac{1}{L_1 + L_2 + L_3}\right)$$
$$\Rightarrow B = \gamma_2 - C.(L_1^2 + L_1.L_2 + L_2^2)$$
$$\Rightarrow A = Y_1 - (B + L_1^2.C).L_1$$

Dans notre dispositif, la thermistance choisie est Temperaturfühler NTC10K -50°C bis +110°C Fühler Sensor Temperatursensor NTC dont les caractéristiques sont les suivantes :

Technische Daten:

- Temperaturbereich: -50 bis +110°C
- Sensordurchmesser: 4mm
- Kabeldurchmesser: 2mm
- komplett wasserdicht
- Genauigkeit: 1%
- ß-Value: 3950

- Kabellänge wählbar

Kennlinie des NTC 10K Sensors

Temperature (°C)	-40	-30	-20	-10	0	10	20	25	30	40	50	60	70	80	90	100	110
Résistance (kΩ)	335,67	176,68	96,79	55,30	32,65	19,90	12,49	10,00	8,06	5,32	3,60	2,49	1,75	1,26	0,92	0,68	0,51

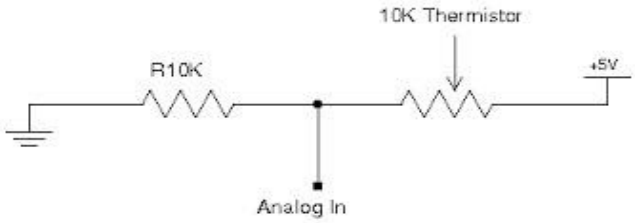
A l'aide de ces caractéristiques, de la loi de Steinhart-Hart, on peut calculer les 3 coefficients A, B, C. Une fois ces 3 coefficients connus, il sera très facile de programmer la formule avec simulink, de la transférer dans le microcontrôleur et d'exécuter le calcul qui nous permettra d'obtenir la température à partir de la variation de résistance.

On obtient : A=0,001110678

B=0,000236962

C=0,000000078.

Cependant, il n'est pas possible pour une entrée analogique en tension d'un microcontrôleur de traiter directement une résistance variable. Il faut transformer cette variation de résistance en une variation de tension. On va utiliser, pour cela, un pont diviseur pour polariser l'entrée analogique de l'Arduino.



Ainsi, le microcontrôleur lira sur son entrée, non pas une variation de résistance, mais une variation de tension (Vanalog) qui s'exprima par la relation :

$$\frac{5}{R10+Rther}=\frac{Vanalog}{R10}$$

On peut remplacer, sans changer le rapport de proportionnalité, les valeurs des tensions par les valeurs numériques correspondantes :

1023 pour 5V,

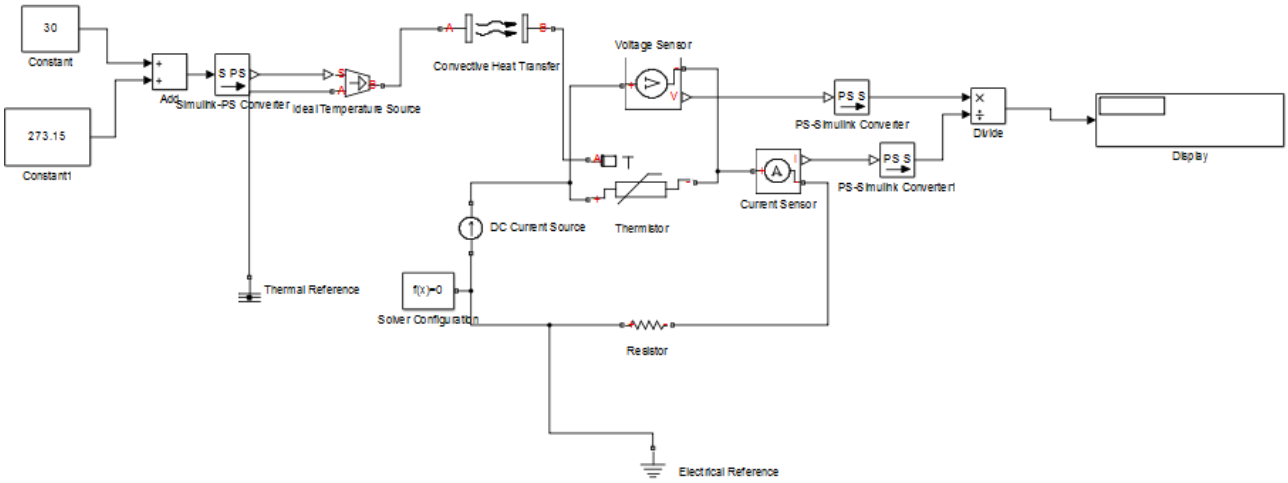
var pour Vanalog.

Var étant la valeur du potentiel variable à l'entrée de la carte exprimée sur 10 bits.

La formule peut alors s'exprimer :

$$\frac{1023*R10}{val}-R10=Rther$$

Une première activité pourra consister, sous Simulink, à modéliser l'ensemble, depuis la grandeur physique mesurée liée à l'environnement jusqu'au branchement de l'entrée Arduino.



On pourra vérifier l'exactitude du modèle par rapport aux données constructeur.

3. Modélisation thermique

L'énergie thermique fournie par le thermoplongeur à l'eau permet l'élévation de la température selon la formule :

$$dW = Mc\, d\,\Theta$$

Ainsi chaque petit accroissement d'énergie apporté élève l'eau d'un petit accroissement de température.

Par ailleurs, la puissance, grandeur physique instantanée s'exprime comme étant la variation d'énergie par unité de temps. Ainsi elle est d'autant plus grande, que l'on peut mobiliser l'énergie en un temps très court. La formule est la suivante :

$$P = \frac{dW}{dt}$$

Hors, pendant que la température de l'eau à l'intérieur du bocal augmente ; le différentiel de température entre l'eau et l'environnement s'accroît également entraînant des pertes thermiques dans les mêmes proportions.

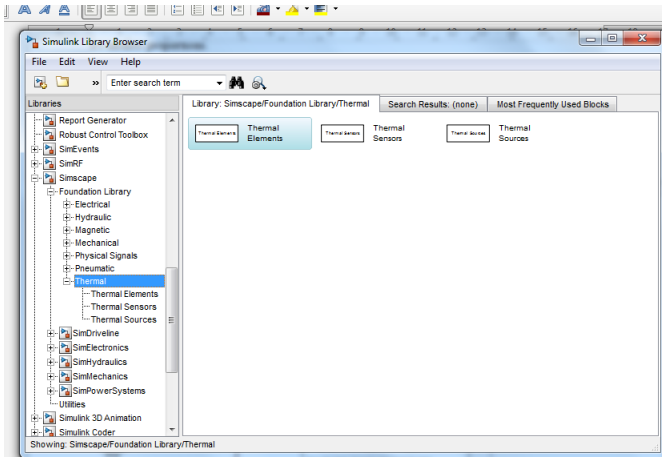
$$Pertes = HS\left(\Theta - \Theta\, ext\right)$$

Avec S en m² et H en W/m²/K

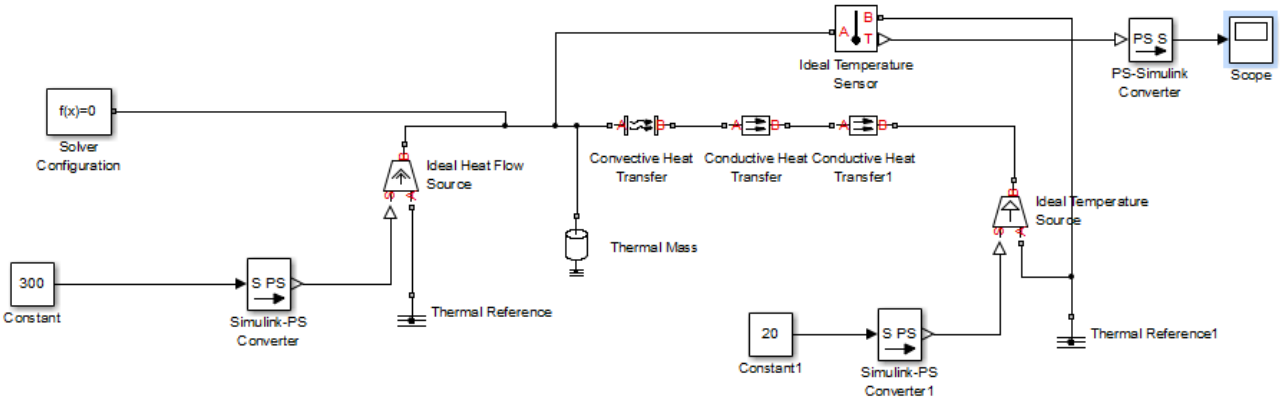
Ce qui donne au final l'expression suivante :

$$P = Mc\frac{d\,\Theta}{dt} + HS\left(\Theta - \Theta\, ext\right)$$

Cette formule peut être directement programmée dans Simulink en utilisant les fonctions mathématiques disponibles, ou bien en utilisant les bibliothèques dédiées dans Simscape :



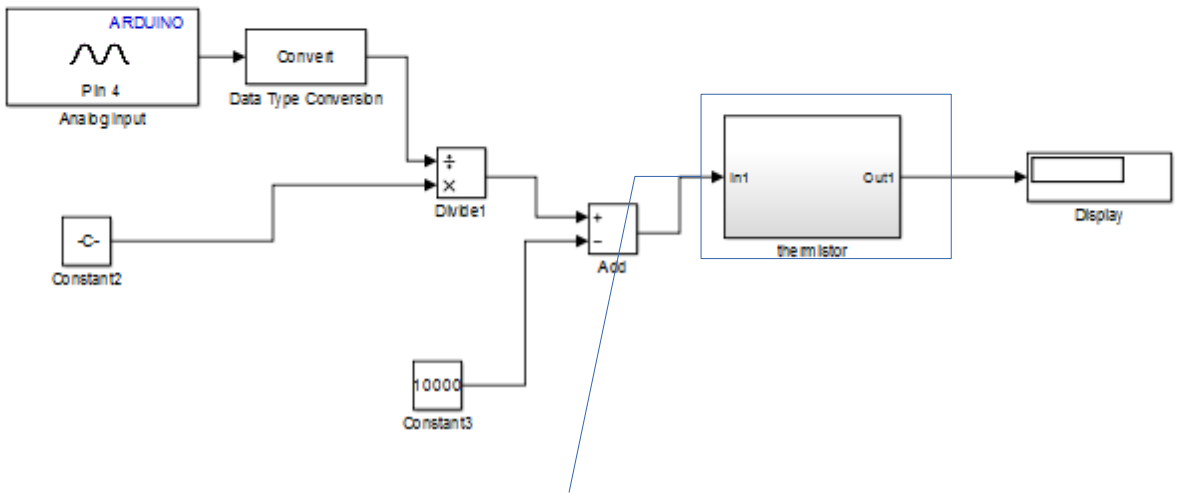
Ci dessous le modèle du bocal rempli d'eau associé à un thermoplongeur de 300 W.

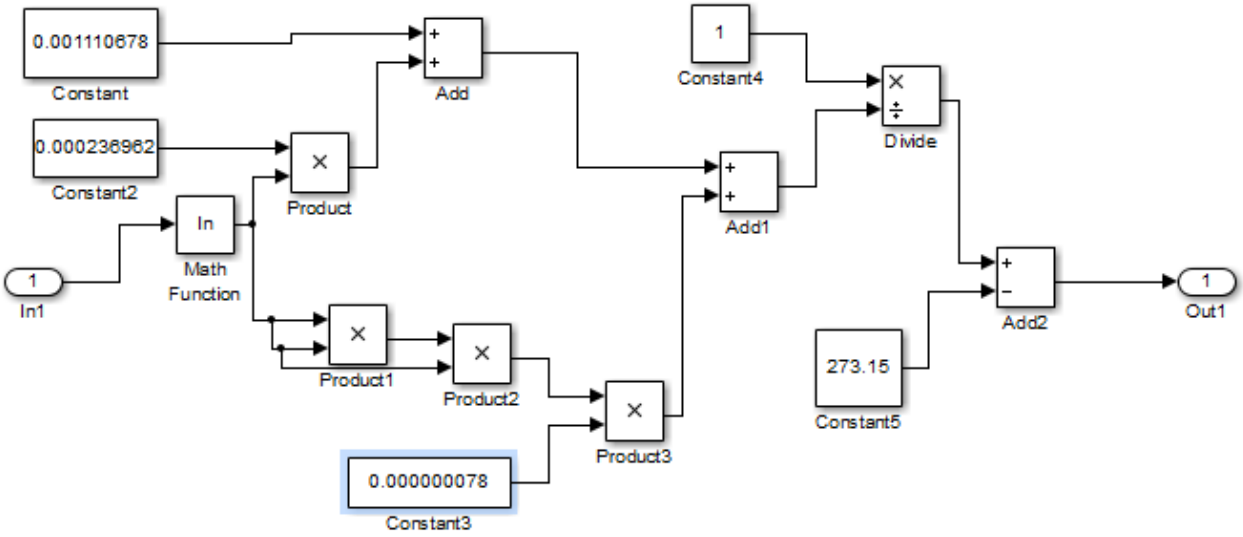


Sur le scope on pourra lire l'évolution de le température de l'eau en fonction du temps.

4. Lecture de la température

Le programme ci-dessous, à partir de la mesure de la tension analogique sur l'entrée 4 de la carte Arduino, permet de remonter à la valeur de la résistance du thermistor pour la traiter ensuite dans le bloc thermistor, siège de la relation Steinhart-Hart.

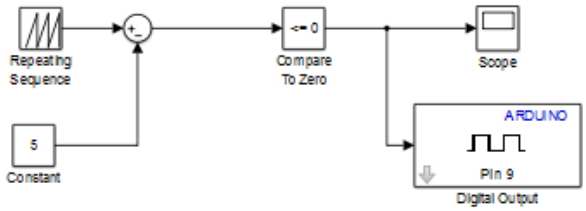




5. Modulation de l'énergie électrique à l'aide du bloc statique

Pour pouvoir moduler l'énergie du thermoplongeur il faut moduler l'énergie électrique que lui fournit l'alimentation. Une commande de type PWM avec une période de 10s fait l'affaire. Seulement, les sorties PWM gérées par Arduino moulinent à des fréquences très élevées, parfaites pour des moteurs DC, mais incompatibles avec les tensions secteurs dont la fréquence vaut 50Hz. Il va donc falloir réaliser nous mêmes la PWM. Il suffit pour cela de comparer la sortie d'un générateur de signaux triangulaires (dents de scie), dont on aura au préalable réglé la période à 10s et l'amplitude à 100 avec une variable comprise entre 0 et 100, image du pourcentage de la puissance à fournir. En sortie du comparateur on aura un signal périodique en créneau de 10s, d'amplitude 1 et de rapport cyclique correspondant à la variable.

Ci dessous le programme pour une puissance transmise de 5 % de 300W.



La sortie 9 de la carte Arduino attaquera la commande du bloc statique.

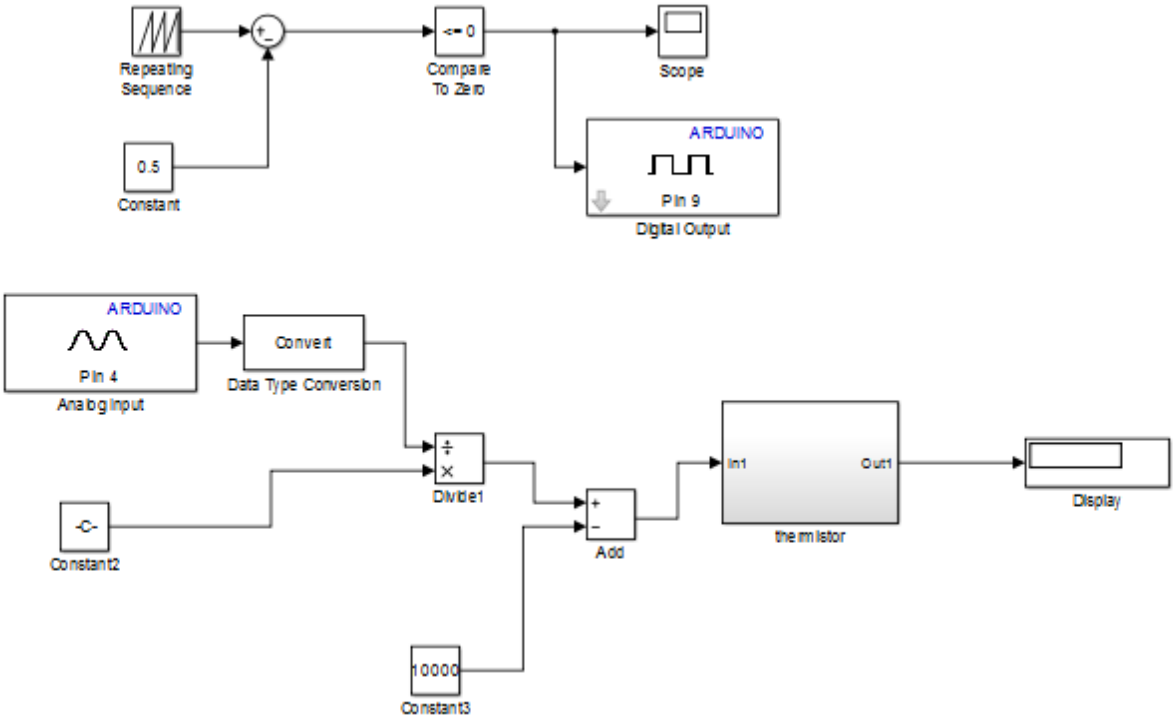
5.1. Essai en boucle ouverte

En réunissant les 2 programmes précédents, on peut procéder à un essai en boucle ouverte. L'idée est de fournir un échelon de puissance au dispositif et d'observer comment celui ci se comporte. On pourra, par exemple, fournir 20 % de la puissance, attendre la stabilisation de la température et relever sa valeur. On en déduira ainsi une estimation der pertes qui pourra nous aider à affiner notre modèle. En effet, lorsque la température est stabilisée, on se trouve dans une situation d'équilibre

thermique et toute la puissance transmise ne sert qu'à lutter contre les pertes.

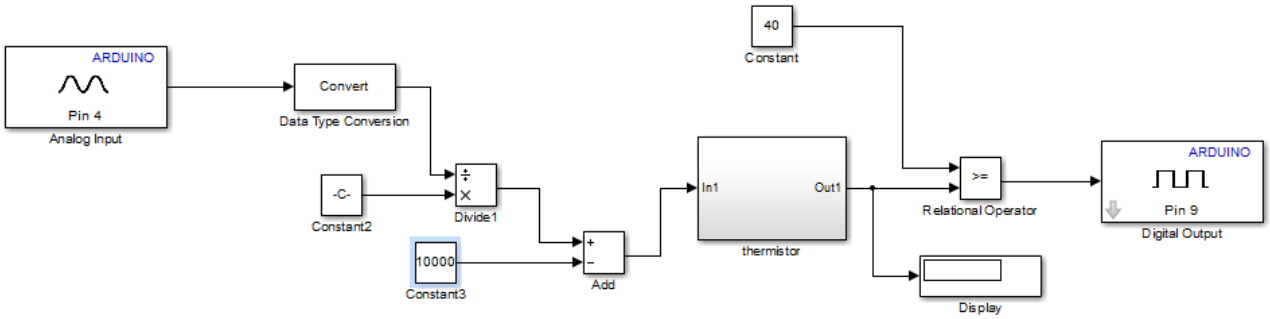
Le terme en $Mc\frac{d\,\Theta}{dt}$

disparaît et il ne subsiste que : $P = HS\left(\Theta - \Theta ext\right)$

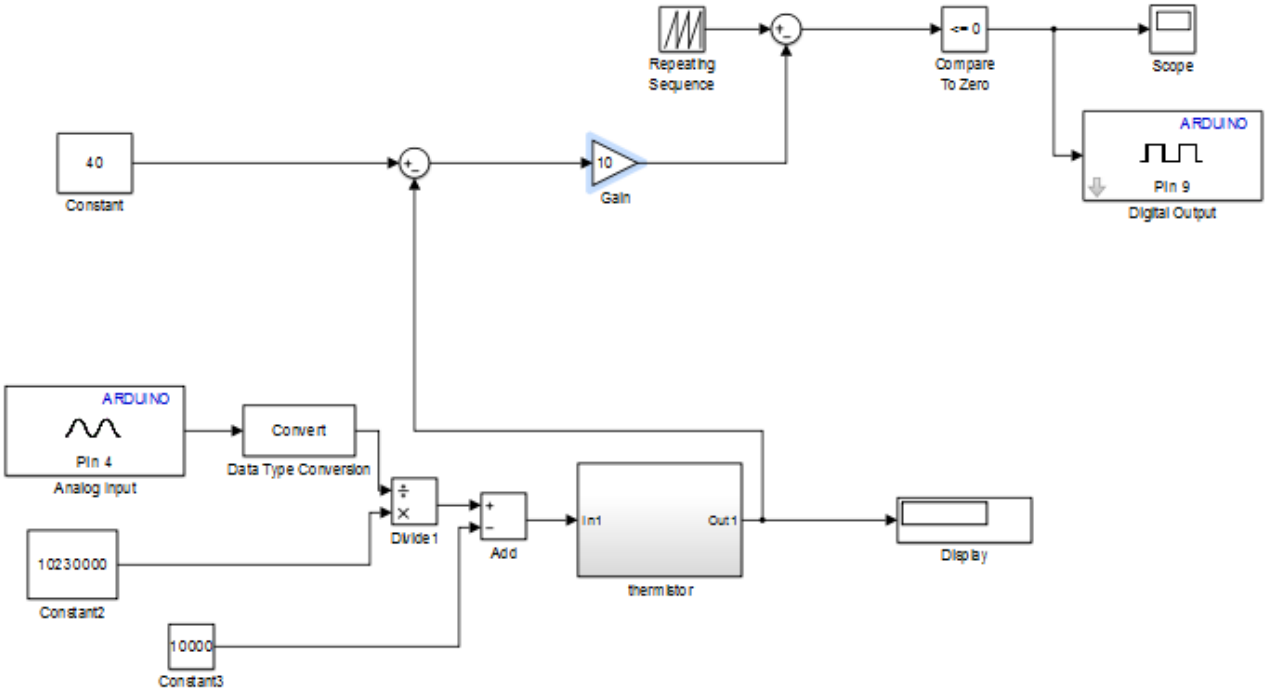


5.2. Essai en boucle fermée de type TOR

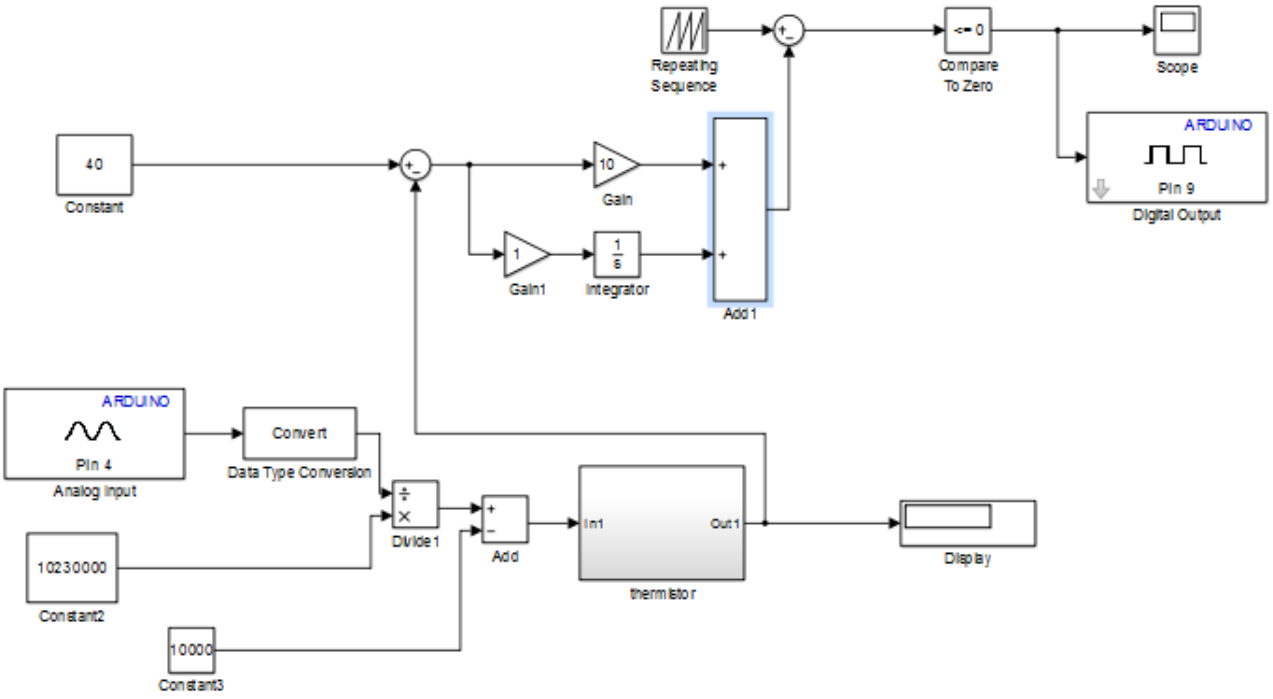
Voici le programme pour réaliser une première régulation de température. Il s'agira, ici, de comparer la consigne (40 °C) à la mesure et de commuter la sortie TOR 9 de l'Arduino tant que la mesure n'atteint pas la consigne. La commande en train d'onde n'est pas utilisée ici. On jugera de l'efficacité d'un tel dispositif.



5.3. Essai en boucle fermée avec correction de type Proportionnel



5.4. Essai en boucle fermée avec correction de type Proportionnel intégral



5.5. Essai en boucle fermée avec correction de type Proportionnel Intégral Dérivé