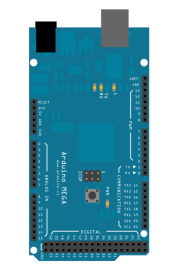
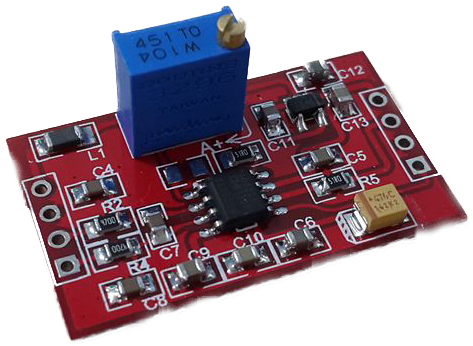
**Choisir et mettre en œuvre une solution  
de mesure d’un effort mécanique**

Problématique :

On souhaite ici mesurer l’effort de portance (dû aux effets aérodynamiques) appliqué sur un véhicule modèle réduit.



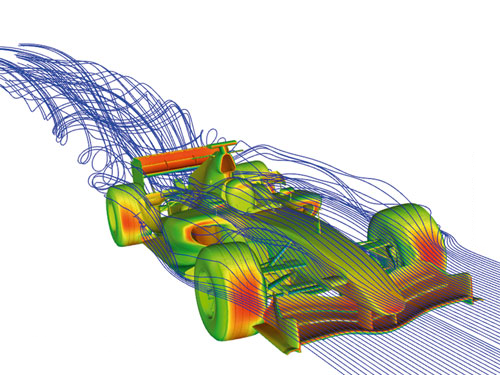
Matériel utile :



* Arduino + Shield Grove + Cable USB



* Capteur d’effort + Carte amplificateur + Câbles Grove
* Masses étalons



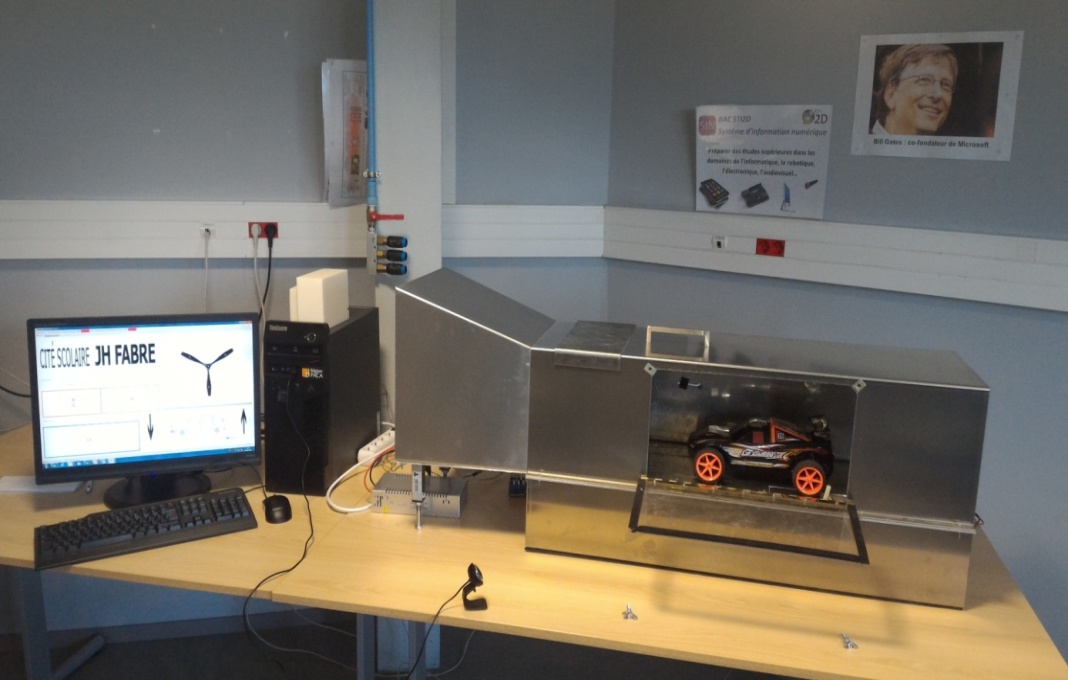
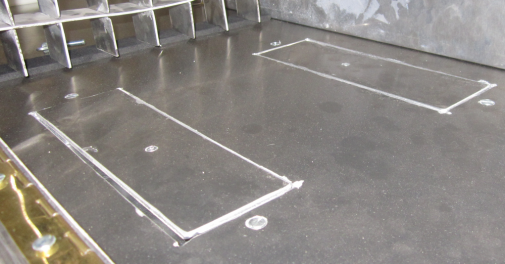
**Poids**

**Portance**

I. Mise en situation et cahier des charges :

L'aérodynamique automobile est l'étude des phénomènes aérodynamiques induits par l'écoulement de l'air autour d'un véhicule automobile en mouvement. La connaissance de ces phénomènes permet de réduire la consommation des véhicules ou d'améliorer leur comportement routier.

Le laboratoire de SI est équipé d’un banc aérodynamique pour modèle réduit. Ce banc dispose, entre autre, de capteurs d’efforts disposés sous chacune des roues afin de mesurer l’effort de portance négative (effort aérodynamique qui a tendance à plaquer une voiture au sol).



Ces capteurs d’efforts sont en fait équivalent des balances que l’on aurait disposés sous chacune des roues.

En absence de flux d’air, ces capteurs sont soumis au poids du véhicule.

En présence d’un flux d’air, ces capteurs, en plus d’être soumis au poids du véhicule sont également soumis à l’effort de portance négative.

Le cahier des charges impose principalement **(1)** :

* **une mesure, sous une roue, dans la gamme 981N** (100g) **à 9025,2N** (920g)
* **une erreur sur la mesure qui soit la plus faible possible**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. L’étude ne porte que sur la mesure de l’effort sous une roue. On considère ici que l’effet cumulé du poids et de l’effort de portance sous la roue sera compris entre 981N et 9025.2N (soit les effets d’une masse étalon d’une valeur comprise entre 100g à 920g que l’on aurait placée sur le capteur d’effort)II. Solutions imaginées à l’issue de la recherche de solutions :

Voici les résultats issus de la recherche de solutions menée par un groupe d’élèves. Lire l’annexe 1.

|  |
| --- |
| *🖒 BONNES PRATIQUES : Avant la mise en place d’un protocole d’expérimentation, il est essentiel d’identifier explicitement la nature de la grandeur à mesurer.* |

* **Capteur d’effort à jauges de déformation sur corps d’épreuve (1) :**

Une fois alimenté par une tension de 5V, ce capteur délivre une tension UCAPT proportionnelle à l’effort à mesurer Fmes ou à la masse m appliquée :

*Schéma d’application envisagé : Caractéristique d’entrée / sortie :*

*Tension U*CAPT *(µV)*

*m (g)*

0

0

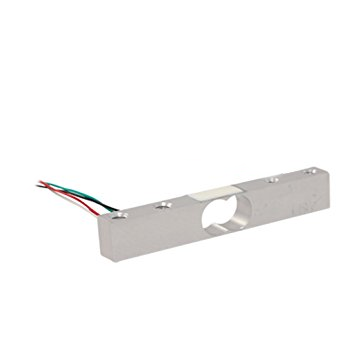
256.4

512.8

0

100

50



*U*CAPT

*Masse*

*+5V*

*Bâti*



*Fmes*

*Masse m*

*Caractéristiques principales du capteur d’effort***(1)** *:*

Sensibilité : **(1/195) mV /g** Etendue de mesure : **0g à 1170g** (limite extrême avant détérioration)

* **Concernant l’organisation de la chaîne d’information trois solutions sont envisagées :**

*Solution 1 :*

**TRAITER**

Conversion A / N 10 bits

AMPLIFICATEUR CARTE ARDUINO

**∩**

Tension *Vref* (V)

#

ACQUERIR  
  
  
  
  
  
  
CAPTEUR D’EFFORT

5V (Alim +)



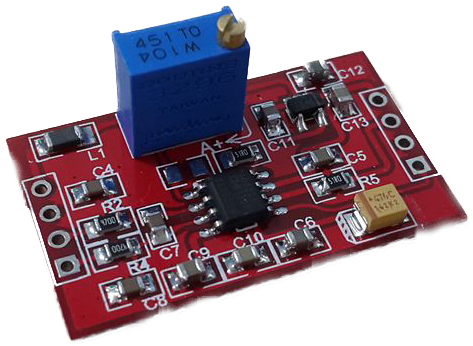
x 830

Donnée numérique *N* image de la température mesurée

Masse à  
mesurer (g)  
*m*

Tension (V)  
*UCAN*

Tension (V)  
*UCAPT*



0V (Alim -)

Dans cette solution, on intercale une fonction multiplication par une constante (amplificateur de tension dont le gain en tension est de 830) entre la sortie du capteur d’effort et l’entrée analogique de la carte Arduino. La pleine échelle VPE du convertisseur A / N est fixée à 5V (valeur par défaut).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Pour simplifier l’étude, on va considérer que :

* l’effort à mesurer (grandeur physique d’entrée à mesurer) varie extrêmement lentement dans le temps (valeur pratiquement continue), seules les propriétés statiques du capteur d’effort et de la chaîne d’acquisition sont alors utiles.
* l’erreur sur la mesure du capteur d’effort est supposée nulle.
* pour éviter les changements d’unité (Newton ⟺ Gramme), l’unité de la grandeur à mesurer sera le gramme.

*Solution 2 :*

ACQUERIR  
  
  
  
  
  
  
CAPTEUR D’EFFORT

**TRAITER**

Conversion A / N 10 bits

AMPLIFICATEUR CARTE ARDUINO

Masse à  
mesurer (g)  
*m*

Tension (V)  
*UCAPT*

Donnée numérique *N* image de la température mesurée

#

**∩**

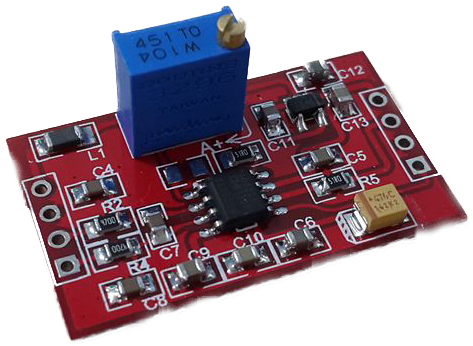
Tension *Vref* (V)

Tension (V)  
*UCAN*

x 1046

0V (Alim -)

5V (Alim +)



Dans cette solution, le gain en tension de l’amplificateur est porté à 1046. La pleine échelle VPE est fixée à 5V.

*Solution 3 :*

La solution 3 reprend exactement la même organisation pour la chaîne d’information que pour la solution 2 mais la pleine échelle VPE du convertisseur A / N est à présent fixée logiciellement à la valeur de 1.1V.

III. Etude théorique de la solution 2 :

Vous devez conduire l’étude théorique de la solution 2 en vue de choisir la solution répondant parfaitement au cahier des charges.

|  |
| --- |
| *🖒 BONNES PRATIQUES : Une étude théorique préliminaire permet de prévoir quantitativement l’ordre de grandeur d’une mesure.* |

🖊 Lire puis compléter l’annexe fourni **(1)** pour la solution 2 afin :

1. D’identifier la nature (analogique, logique et numérique) en divers points de la chaine d’information ;
2. Compléter le schéma bloc en donnant les relations liant :
   1. UCAPT et masse à mesurer m ;
   2. UCAN et UCAPT ;
   3. N et la tension UCAN puis N et la masse mmes ;
   4. mmes et l’effort Fmes en Newton.
3. De déterminer les valeurs prises par les tensions UCAPT, UCAN et par la donnée numérique N en fonction de la masse m mesurée ;
4. De critiquer la pertinence de la solution proposée.

🖊 Compléter le tableau de choix de solution de la page suivante.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Se référer à l’annexe 2.

IV. Choix d’une solution :

|  |
| --- |
| *🖒 BONNES PRATIQUES : Une solution n’est retenue que si elle respecte tous les critères quantitatifs initialement attendus. Le meilleur moyen de vérifier si une solution respecte le cahier des charges initial est d’élaborer un tableau listant chacune des solutions possibles et les critères attendus.* |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Choix d’une solution pour la fonction ACQUERIR*** | | | |
| ***Critères de choix*** | | | **Solution : Capteur d’effort** |
| ***Etendue de mesure*** | Etendue de mesure du capteur choisi | |  |
| Gamme de d’efforts de portance à mesurer d’après le cahier des charges | | 100g à 920g |
| Solution envisageable ou non | | □ Envisageable □ Non envisageable |
| ***Erreur sur la mesure*** | Erreur sur la mesure du capteur choisi | |  |
| Erreur sur la mesure souhaitée d’après le cahier des charges | | Erreur la plus faible possible |
| Solution envisageable ou non | | □ Envisageable □ Non envisageable |
| ***Choix de la solution et justification*** | | La solution choisie respecte tous les critères du cahier des charges. Le choix du capteur d’effort est définitivement validé. | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Choix d’une solution pour la fonction TRAITER*** | | | | | |
| ***Critères de choix*** | | | **Solution 1** | **Solution 2** | **Solution 3** |
| ***Etendue de mesure*** | Gamme de valeurs prises par N | | 0 à 1023 | 0 à 1023 | 0 à 1023 |
| Relation de m (en g) en fonction de N | | m = (N x 5 / 1024) / 830 x 195000 |  |  |
| Etendue de mesure de la solution (en g) | | 0 à 1173g |  | 0 à 204g |
| Gamme de d’efforts de portance à mesurer d’après le cahier des charges | |  |  |  |
| Solution envisageable ou non | | ☑ Envisageable □ Non envisageable | □ Envisageable □ Non envisageable | □ Envisageable □ Non envisageable |
| ***Erreur sur la mesure*** | Plus petite variation possible de N | | 1 | 1 | 1 |
| Relation de m (en g) en fonction de N | |  |  |  |
| Plus petite variation d’effort de portance mesurable (en g) avec la solution (erreur sur la mesure) | | 1,14g |  | 0,20g |
| Solution envisageable ou non | | □ Envisageable ☑ Non envisageable | □ Envisageable □ Non envisageable | ☑ Envisageable □ Non envisageable |
| ***Choix de la solution et justification*** | |  | | | |

V. Modélisation de la solution choisie

|  |
| --- |
| *🖒 BONNES PRATIQUES : Dans un projet, la modélisation permet de valider une solution et d’anticiper les problèmes avant sa mise en œuvre réelle.* |

🖊 Saisir, sur Matlab / Simulink le modèle causal correspondant à la chaine d’acquisition de la solution choisie pour la mesure de l’effort :

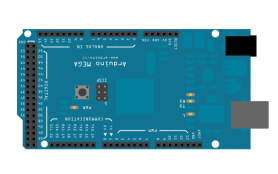
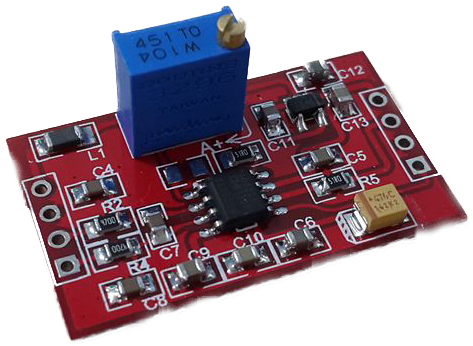
*Simulink>  
Sinks>  
Display*

*Simulink>  
Commonly>Constant*

*Simulink>  
Commonly>Gain*

*Simulink>  
Math Operations>  
Rounding Function*

*Composants   
modélisés*

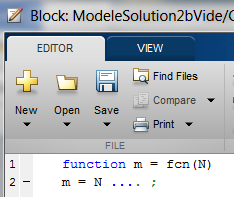


*Composants réels*

🖊 Faire une capture d’écran de votre modèle causal :

🖊 Ajouter au modèle précédent une fonction Matlab afin de calculer la valeur de masse mesurée mMES à partir de la valeur N :

*Simulink>  
User-Defined Functions*



*Pour saisir le calcul à effectuer il faut double-cliquer sur le bloc User-Defined Functions puis saisir les informations suivantes (remplacer les … par le calcul à effectuer) :*

🖊 Faire une capture d’écran de votre modèle et du contenu du bloc « fcn » :

🖊 Dans la constante Masse, saisir les différentes valeurs de masse en g sous la forme d’un tableau :  
[0 50 100 200 300 etc…]



🖊 Exécuter le modèle en cliquant sur l’icône afin de compléter le tableau ci-dessous. Calculer l’erreur sur la mesure issue du modèle :



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Masse à mesurer m en g* | *U*CAPT *en mV* | *U*CAN *en V* | *N sans unité* | *Masse mesurée mMES en g* | *Erreur (valeur absolue) sur la mesure en g* |
| 0g |  |  |  |  |  |
| 50g |  |  |  |  |  |
| 100g |  |  |  |  |  |
| 200g |  |  |  |  |  |
| 300g |  |  |  |  |  |
| 600g |  |  |  |  |  |
| 700g |  |  |  |  |  |
| 800g |  |  |  |  |  |
| 900g |  |  |  |  |  |
| 930g |  |  |  |  |  |

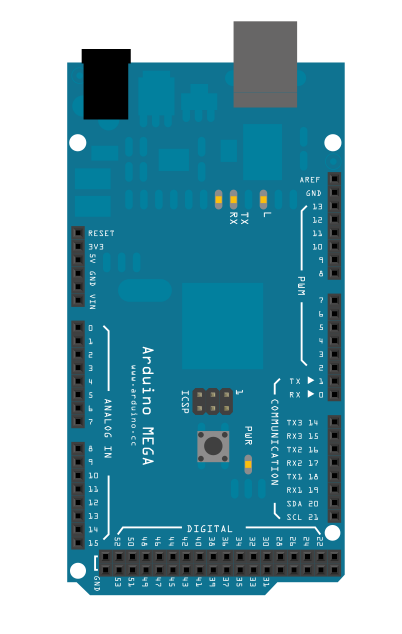
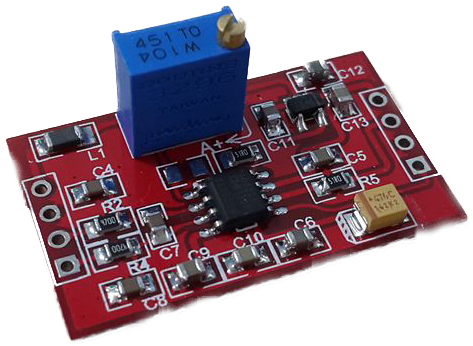
🖊 L’erreur maximale issue du modèle en g est-elle cohérente avec l’étude théorique précédente ? Le modèle est-il en adéquation avec l’analyse théorique ?

VI. Expérimentation de la solution et évaluation des écarts avec le modèle

1. *Mise en œuvre matérielle :*

Voici le schéma d’implantation matériel de la solution que l’on souhaite mettre en œuvre afin de **confronter les résultats issus du modèle et issus du rée**l :

***Modèle créé précédemment :***



*U*CAPT



*Bâti*



*Masse m*

*Transmission de la donnée N issue de la conversion analogique / numérique via le port USB*

*U*CAN

*Amplificateur (G = 1046)* **(1)**

*Capteur d’effort*

*Carte Arduino*



***Réel (schéma d’implantation) :***

🖑 A partir du schéma d’implantation de solution ci-dessus, mettre en œuvre la solution choisie pour la mesure de l’effort. **Faire appel au professeur**.

**NE PAS CONNECTER LE CABLE USB AU PC !**

1. *Mise en œuvre logicielle :*

Vous savez que Matlab / Simulink permet de **modéliser une solution**. Mais pas seulement…

Matlab / Simulink permet également d’**expérimenter une solution** !

Matlab / Simulink peut par exemple communiquer avec une carte Arduino, connectée au port USB du PC, afin de très facilement :

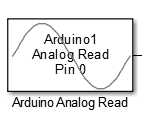
* *Manipuler les informations numériques issues de la carte Arduino*
* *D’afficher les résultats d’une expérimentation à l’aide des composants Simulink habituel : Scope, etc…*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

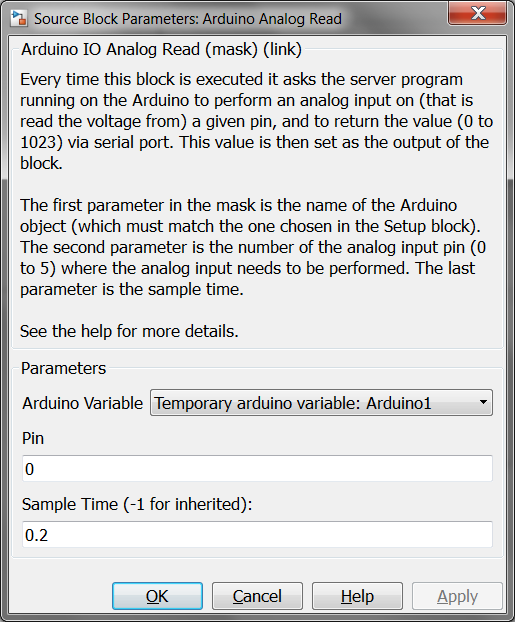
1. A noter que le réglage du gain de l’amplificateur se fait via un potentiomètre. Le professeur a déjà réglé le gain à la valeur souhaitée.

*Arduino IO Library Simulink*

1. *Saisie du schéma d’expérimentation sur Matlab / Simulink :*



Le bloc « Arduino AnalogRead » présent dans la librairie « Arduino IO Library » permet de récupérer, via la liaison USB, le résultat de la conversion analogique / numérique réalisée par la carte Arduino :



Dans les propriétés du bloc « Arduino AnalogRead » il est possible de :

* *Définir la broche analogique que l’on souhaite utiliser (broche sur laquelle on a connecté un capteur par exemple), ici : broche 0*
* *Définir la période d’échantillonnage (durée entre deux récupérations de la donnée issue de la conversion analogique numérique), ici : 200ms*

*Arduino IO Library Simulink*



Le bloc « Real-Time Pacer » présent dans la librairie « Arduino IO Library » permet de définir la vitesse d’exécution du schéma d’expérimentation (Speedup = 1 : Fonctionnement temps réel).



Le bloc « Arduino IO Setup » présent dans la librairie « Arduino IO Library » permet le port de communication associé à la carte Arduino connectée au PC.

Le schéma d’expérimentation Matlab / Simulink ci-dessous permet par exemple :

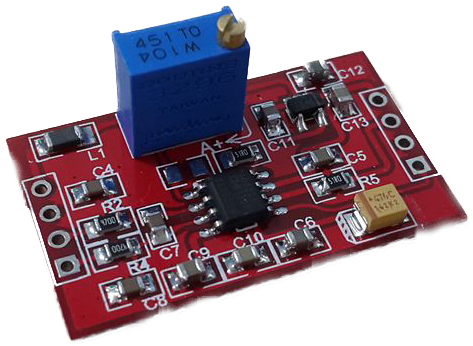
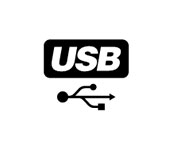
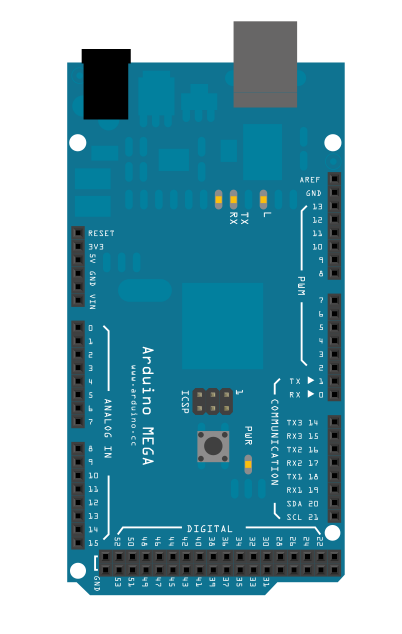
1. de récupérer la donnée N issues de la conversion analogique / numérique (broche 0, toutes les 200ms)
2. de calculer la valeur de la masse mesurée mmes (il faut, bien  
   évidemment, pour cela compléter correctement le bloc « fcn »).



1. *Schéma d’expérimentation (aspect matériel et logiciel) permettant la mesure de l’effort*

*Transmission de la donnée N issue de la conversion analogique / numérique via le port USB*

Voici le récapitulatif complet de l’expérimentation permettant la mesure de l’effort :



*U*CAN

*Masse m*



*Ordinateur  
avec  
Matlab / Simulink*

*U*CAPT

*Capteur d’effort*

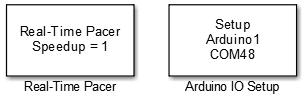
*Amplificateur*

*Carte Arduino*

*Bâti*

*Voltmètre 2*

*Voltmètre 1*



*Schéma Simulink d’expérimentation*

🖊 Compléter le schéma de la page précédente en connectant les voltmètres de manière à mesurer, au signe, prés :

* la tension UCAPT avec le voltmètre 1 ;
* la tension UCAN avec le voltmètre 2.

🖑 Saisir le schéma d’expérimentation Matlab / Simulink de la page précédente.

🖑 Compléter le bloc « fcn » de façon à calculer la valeur de la masse mesurée mmes à partir de la valeur de N (convertir N en type single en écrivant : single(N)). Faire une capture des instructions saisies dans le bloc « fcn » :

🖑 Enregistrer le schéma d’expérimentation Matlab / Simulink sous le nom « MesureMasse.slx » dans votre dossier personnel.

1. *Mesure des grandeurs en divers points de la chaîne d’acquisition*

On souhaite ici mesurer les grandeurs électriques en divers points de la chaîne d’acquisition afin de valider son bon fonctionnement.

**A REALISER AVEC LE PROFESSEUR !!**

*Relevé de la tension UCAPT pour différentes valeurs de masse :*

🖑 Connecter le voltmètre afin de mesurer la tension UCAPT. Puis connecter la carte Arduino au port USB afin d’alimenter le capteur d’effort, l’amplificateur ainsi que la carte Arduino.

🖊 Compléter la colonne UCAPT du tableau ci-dessous. Déconnecter la carte Arduino du port USB.

*Relevé de la tension UCAN pour différentes valeurs de masse :*

🖑 Connecter le voltmètre afin de mesurer la tension UCAN. Puis connecter la carte Arduino au port USB.

🖊 Compléter les colonnes UCAN et Gain = CCAN / UCAPT du tableau ci-dessous. Déconnecter la carte Arduino.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Masse à mesurer m en g* | *U*CAPT *en mV* | *U*CAN *en V* | *Gain* |
|  | 0g |  |  |  |
|  | 50g |  |  |  |
|  | 100g |  |  |  |
|  | 200g |  |  |  |
|  | 300g |  |  |  |
|  | 400g |  |  |  |
|  | 450g |  |  |  |
| Valeur (déduite) maximale mesurable obtenu en appuyant progressivement sur le capteur |  |  |  |  |
|  | 920g | 4.7 |  |  |

🖊 Sur Excel tracer les caractéristiques d’entrée/sortie du :

* capteur UCAPT = f(m)
* amplificateur UCAN = f(UCAPT)

UCAPT(mV) = f(m(g)) UCAN(V)= f(UCAPT(mV))

🖊 Commenter les tracés précédents :

UCAPT(mV) = f(m(g)) Sensibilité ? Valeur à l’origine ?

|  |
| --- |
|  |

UCAN(V)= f(UCAPT(mV)) Gain ? Valeur maximale ?

*Relevé de la donnée numérique N issue de la conversion analogique / numérique :*

🖑 Ouvrir le schéma d’expérimentation Matlab / Simulink « MesureMasse.slx » et modifier le bloc « fcn » (calcul de la valeur de la masse mesurée mmes à partir de la valeur de N) de façon à prendre en compte la tension présente en sortie de l’amplificateur (par exemple ici -0.503V) en absence de masse (m = 0g) :



🖑 Connecter la carte Arduino au port USB afin de visualiser la valeur de la donnée N et de la masse mesurée mmes par la chaine d’acquisition.

🖑 Exécuter le schéma d’expérimentation en cliquant sur l’icône



🖑 Compléter le tableau ci-dessous :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Masse à mesurer m en g* | *U*CAN *en V* | *N sans unité* | *Masse mesurée mMES en g* | *Erreur (valeur absolue) sur la mesure en g* |
|  | 0g |  |  |  |  |
|  | 50g |  |  |  |
|  | 100g |  |  |  |  |
|  | 200g |  |  |  |  |
|  | 300g |  |  |  |  |
|  | 400g |  |  |  |  |
|  | 450g |  |  |  |  |
| Valeur (déduite) maximale mesurable obtenu en appuyant progressivement sur le capteur |  |  |  |  |  |
|  | 920g |  |  |  |  |

1. *Analyse des écarts et préconisation de modifications de la chaîne d’acquisition*

|  |  |  |  |
| --- | --- | --- | --- |
| *Résultats de la solution de mesure de masse :* ***Etendue de mesure*** | | | |
| ***Etendue de mesure*** | Etendue de mesure de la solution réelle | |  |
| Etendue de mesure prévue par l’étude théorique | |  |
| ***Solution envisageable ou non en l’état*** | | □ Envisageable □ Non envisageable en l’état |
| ***Analyse des résultats de mesure*** | |  | |
| ***Modification envisagée*** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| *Résultats de la solution de mesure de masse :* ***Erreur sur la mesure*** | | | |
| ***Erreur sur la mesure*** | Erreur maximale commise sur la mesure avec la solution choisie | |  |
| Erreur sur la mesure prévue par l’étude théorique | |  |
| Erreur sur la mesure souhaitée d’après le cahier des charges | |  |
| ***Solution envisageable ou non*** | | □ Envisageable □ Non envisageable en l’état |
| ***Analyse des résultats de mesure*** | |  | |