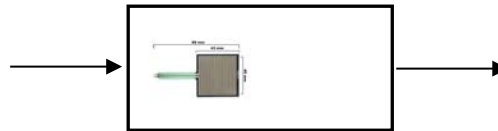


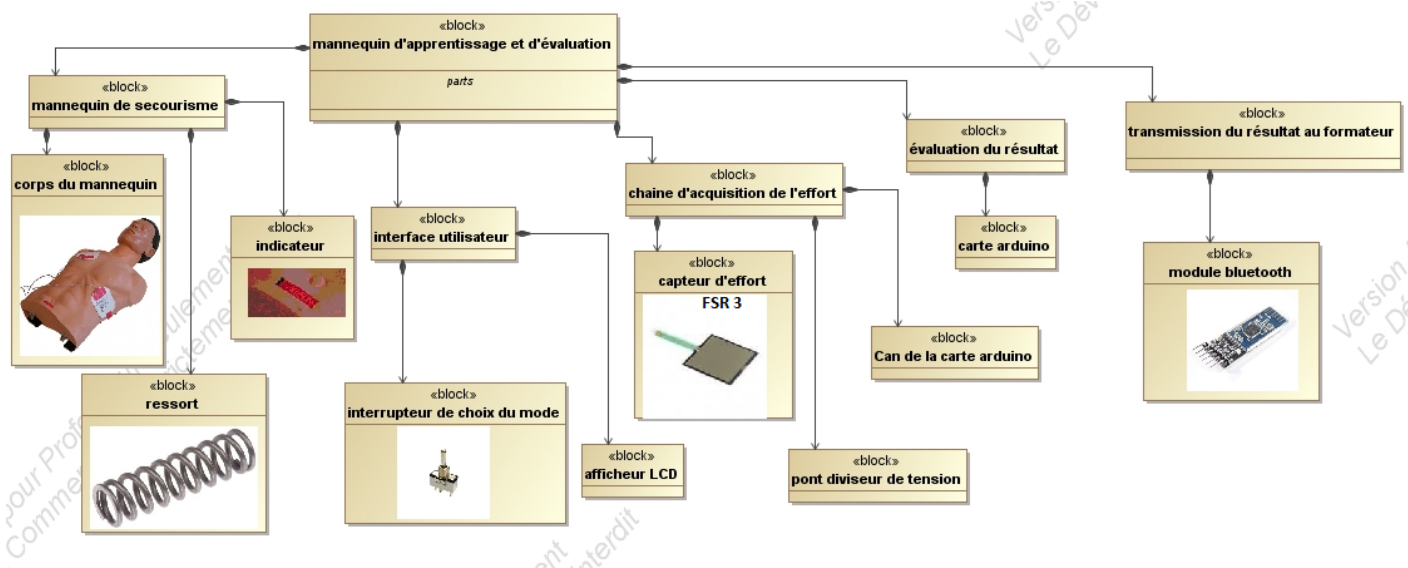
## Activité 1 EQUIPE VERTE élève 5: Comment mesurer un effort ?

- Rechercher dans le diagramme des blocs ci-dessous, le nom du composant chargé de traduire l'effort en variation de résistance. Consulter sa documentation technique dans le répertoire « ressources » puis compléter le schéma ci-dessous :

Grandeur physique à mesurer :



Grandeur électrique en sortie :

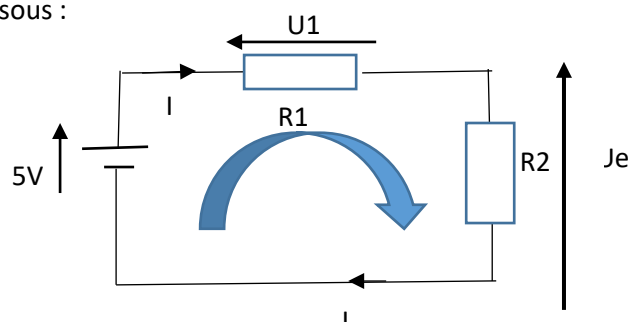


Nous devons maintenant connaître la variation de résistance en sortie du capteur correspondant à un effort correctement dosé.

- Proposer un protocole expérimental permettant de tracer la courbe de réponse du capteur  $R_{\text{cpt}}=f(F)$ . Tracer cette courbe. Comment évolue  $R_{\text{cpt}}$  quand l'effort augmente ?

## Comment traduire la variation de résistance en sortie du capteur d'effort par une variation de tension ?

Nous avons vu que l'effort appliqué sur le thorax du mannequin était traduit par le capteur FSR3 en une variation de résistance. Nous avons déterminé la plage de variation de cette résistance pour un effort correct. Pour qu'une carte à microcontrôleur puisse vérifier que l'effort est correct nous devons traduire la variation de résistance en sortie du capteur FSR3 en une variation de tension. Pour cela, nous allons utiliser une structure appelée « pont diviseur de tension » dont le schéma est donné ci-dessous :

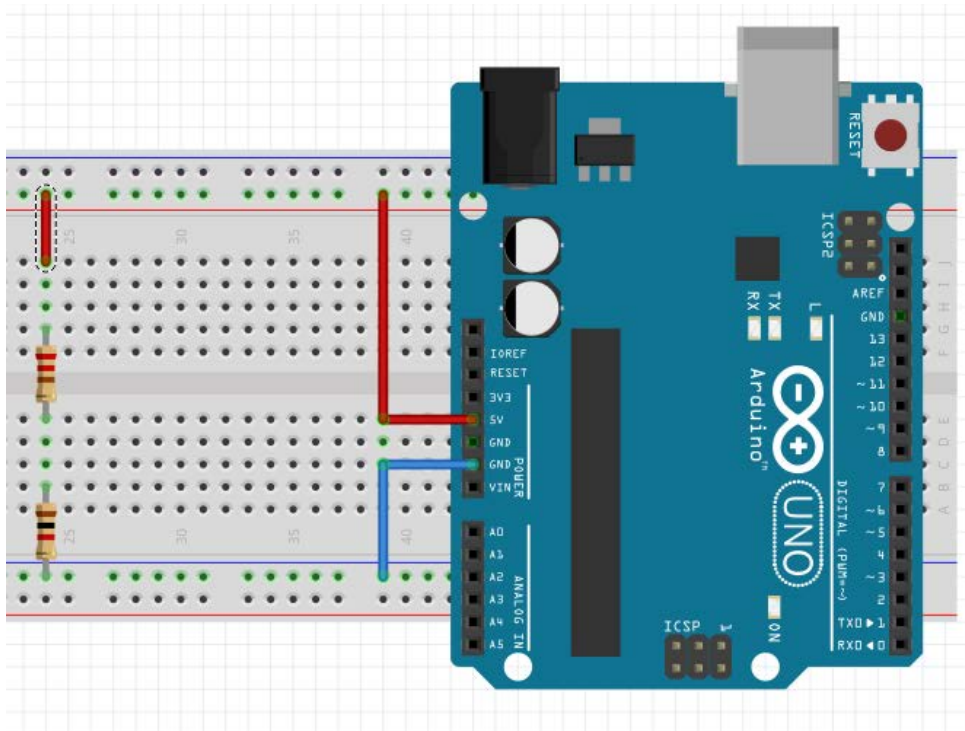


Remarque : dans le mannequin, la résistance R1 sera remplacée par le capteur d'effort.

## CHALLENGE MANNEQUIN

La tension  $U_e$  aux bornes de la résistance  $R_2$  est donnée par la relation :  $U_e = \frac{5 \cdot R_2}{R_1 + R_2}$

- Indiquer comment évolue  $U_e$  quand la résistance  $R_1$  diminue.
- En prenant deux valeurs de résistances de votre choix, comparer la valeur de  $U_e$  mesurée avec la valeur théorique calculée grâce à la relation ci-dessus. Comparer les deux valeurs et justifier les éventuels écarts.

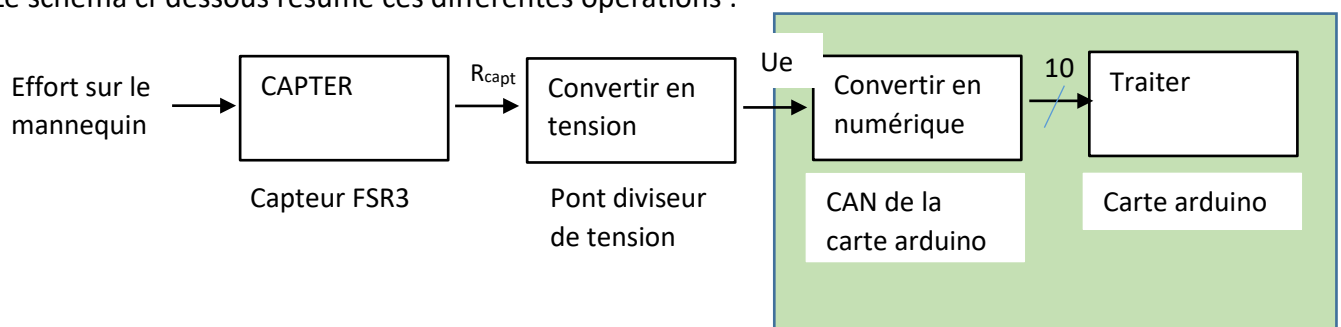


Valeurs de résistances R1= R2=	Valeur théorique de $U_e$ (calcul)	Valeur expérimentale de $U_e$	Ecart entre la valeur théorique et la valeur expérimentale

### Comment le microcontrôleur peut lire une tension ?

Un microcontrôleur exécutera les instructions du programme que vous réaliserez. Ce microcontrôleur ne comprend que le langage binaire composé uniquement de 0 et de 1 (absence ou présence de tension). Il faudra donc convertir la tension  $U_e$  image de l'effort en valeur numérique (binaire) pour que celle-ci puisse être analysée par le microcontrôleur.

Le schéma ci-dessous résume ces différentes opérations :



## CHALLENGE MANNEQUIN

- ✚ La conversion analogique numérique est réalisée sur 10 bits. Ecrire en binaire la plus petite puis la plus grande valeur que l'on peut obtenir en sortie du CAN. Compléter la colonne de droite du tableau ci-dessous :

Tension en entrée du CAN	Valeur théoriques numérique en sortie du CAN	Valeur binaire sur 10 bits en sortie du CAN
0V	Nmin=	Val min=
5V	Nmax=	Val max=
$U_e =$ (valeur expérimentale mesurée précédemment)		

- ✚ Sous Windows, ouvrir la calculatrice, choisir le mode « programmeur » et réaliser la conversion binaire/décimale vous permettant de convertir ces valeurs en décimal. Compléter la colonne « valeur numérique en sortie »
- ✚ Sachant qu'il y a proportionnalité entre les 2 premières colonnes du tableau, compléter la ligne suivante. (Vous utiliserez à nouveau la calculatrice en mode programmeur pour la colonne de droite).

Le  $\mu C$  de la carte arduino comprend 6 convertisseurs analogique numérique (CAN) accessibles sur les entrées A0..A5 de la carte Arduino. Il va donc falloir réaliser un programme sous Arduino pour mettre en œuvre ces CAN.

Le programme devra suivre l'algorithme suivant :

*Constante entière  $U_e$  égale A0*

*variable valeur\_num\_sortie en entier*

*debut*

*Faire*

*valeur\_num\_sortie égale conversion de  $U_e$*

*afficher valeur\_num\_sortie*

*tant que toujours*

*fin*

- ✚ Positionner sur le schéma de la page précédente, la variable « valeur\_num\_sortie ».

## CHALLENGE MANNEQUIN

L'algorithme ci-dessous peut être traduit en langage de programmation C pour Arduino de la façon suivante :

```
const int  tension_en_entree=A0 ;

int valeur_num_sortie ;

void setup()  //fonction de configuration
{
  Serial.begin(9600) ; //configuration du débit pour affichage sur écran
}

void loop()  //fonction exécutée en boucle
{
  valeur_num_sortie=analogRead(tension_en_entree) ;
  Serial.println(valeur_num_sortie) ;
  delay(1000) ;
}
```

- Ouvrir le programme Arduino sur le bureau. Créer un nouveau fichier et copier le programme ci-dessus. Puis compiler votre programme (transformer le en binaire) et téléverser votre programme.

Lancer le moniteur série et visualiser les valeurs de *valeur\_num\_sortie*.

- Compléter votre montage en connectant la broche A0 de la carte arduino puis compléter le tableau ci-dessous. Conclure sur les éventuels écarts.

Tension en entrée du CAN	Valeur théoriques numérique en sortie du CAN	Valeur binaire sur 10 bits en sortie du CAN
0V	Nmin=	Val min=
5V	Nmax=	Val max=
$U_e =$		