

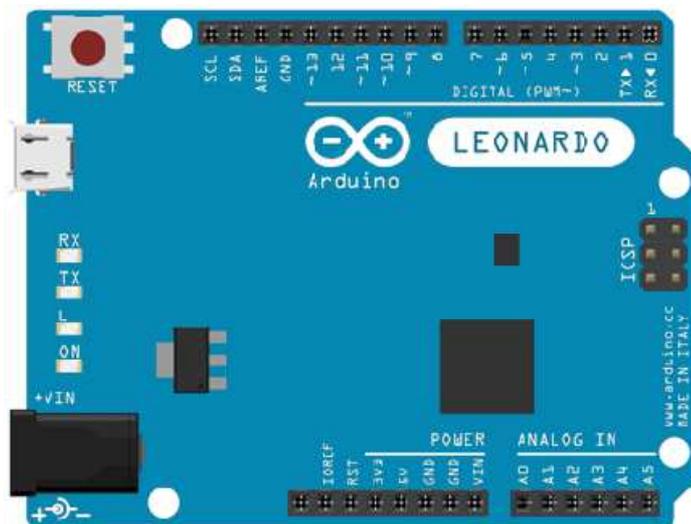
IX. Décodeur BCD et afficheur 7 segments

Plan de câblage et programme pour la réalisation d'un compte à rebours sur un afficheur 7 segments câblé à l'aide d'un décodeur BCD et d'un Arduino Léonardo. L'afficheur 7 segments utilisé est le même que pour l'exercice précédent, nous utiliserons donc les mêmes résistances.

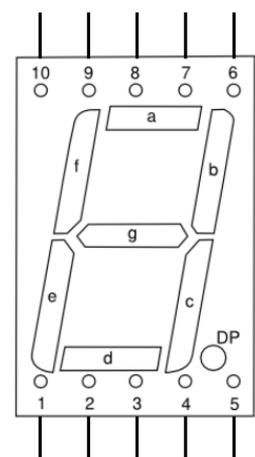
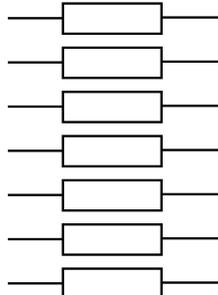
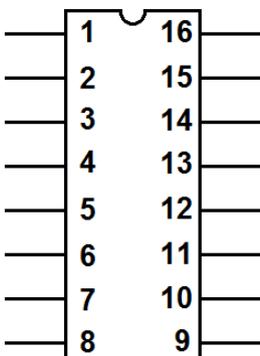
1) A l'aide des datasheets « 5161AS » et « CD4543BE », **complète le schéma de câblage de l'afficheur 7 segments sur le décodeur BCD, et du décodeur BCD sur l'Arduino** (utilise des couleurs différentes pour une lecture plus rapide, et trace les traits pour qu'ils se coupent le moins possible), avec :

Bit	Broche Arduino
bit_a (2^0)	A2
bit_b (2^1)	A4
bit_c (2^2)	A5
bit_d (2^3)	A3

Attention : aucune broche ne doit rester non câblée. A toi de chercher à quoi servent les broches numéros : 1, 6, 7, 8 et 16 et les relier soit au 0V (en noir) soit au 5V (en rouge).



$R = \dots\dots\dots \Omega$



2) Réalisation du programme :

a) **Enregistre** ce programme dans tes documents sous le nom « 09_decodeur_BCD »

b) **Avant le setup () :**

Assigner chaque entrée du décodeur à sa broche (s'aider du tableau de la question 1).

c) **Le setup () :**

Indiquer que les broches seront utilisées en sorties (pour piloter le décodeur BCD).

d) Le loop ()

Réaliser le programme principal qui est un compte à rebours de 9 à 0 tournant en boucle, avec $-1 = 1$ seconde. Pour cet exercice nous allons utiliser une boucle « do...while » et une variable « n » de type « char » :

- Créer la variable « n » avec une valeur initiale de 9 : premier chiffre à afficher dans notre compte à rebours.

- Faire...:

- Appeler un sous programme « dizaine » avec comme paramètre la valeur « n ». (ce sous-programme va convertir la valeur « n » en nombre binaire, pour piloter le décodeur BCD)

- Puis attendre une seconde.

...Tant que « n » sera différent de « -1 » (donc une valeur comprise entre 9 et 0)

e) Sous-programme « dizaine »

Il faut maintenant convertir « n » en un nombre binaire. Sachant que :

$$\text{bit_a} = 2^0 = 1$$

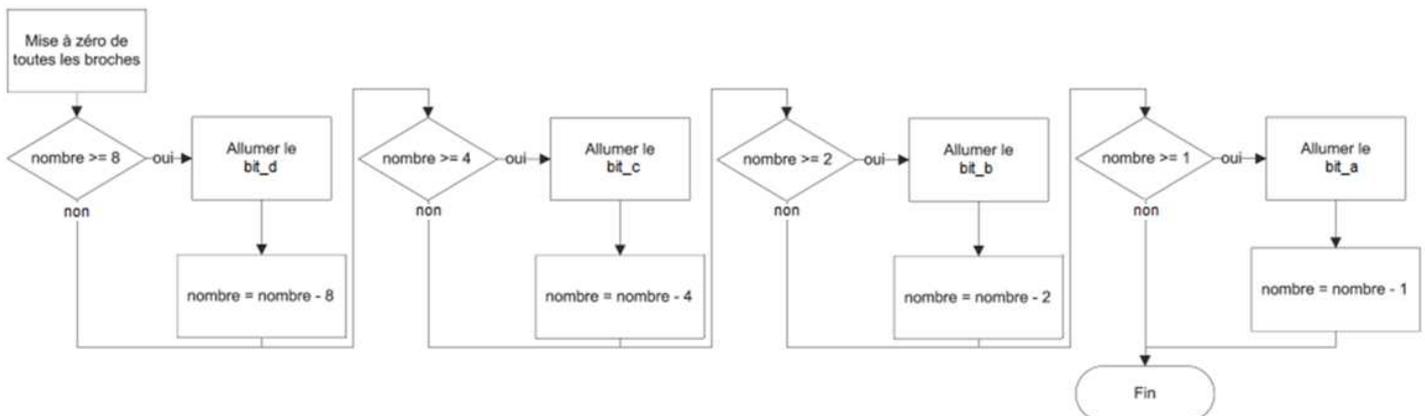
$$\text{bit_b} = 2^1 = 2$$

$$\text{bit_c} = 2^2 = 4$$

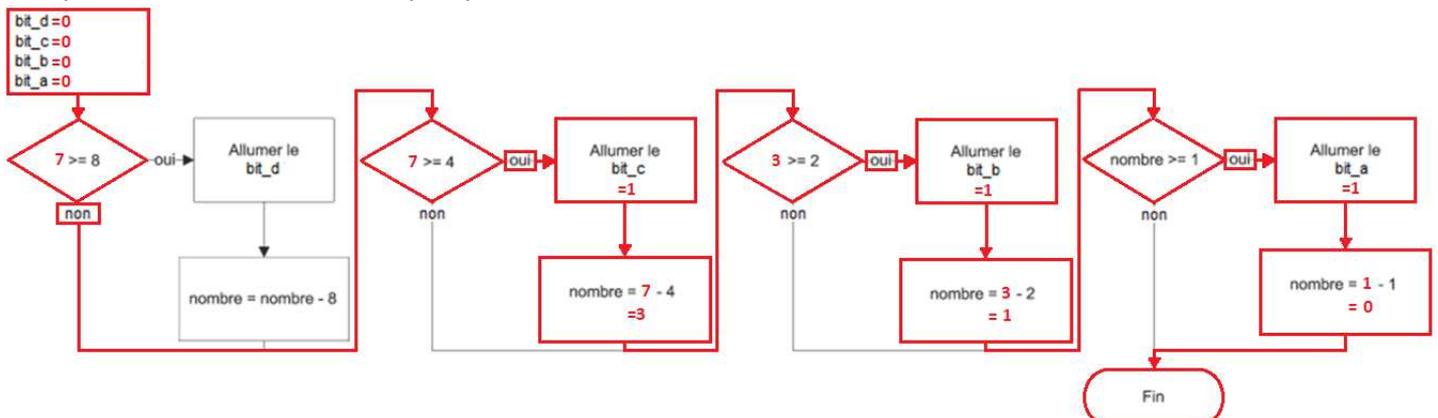
$$\text{bit_d} = 2^3 = 8$$

La valeur « n » reçue dans la fonction, sera enregistrée dans une variable « nombre » de type « char ».

On met tous les bits du décodeur à 0. Ensuite nous allons comparer la valeur de « nombre » aux valeurs des différents bits, pour en mettre certains à 1. Il faut commencer par comparer la valeur de "nombre" avec la valeur du bit le plus fort (ici bit_d, soit 8). Si elle est supérieure ou égale, il faut passer le bit à 1 et enlever la valeur de ce bit à "nombre". Sinon on ne change pas la valeur de "nombre". Puis il faut faire de même avec le bit inférieur, jusqu'au bit le plus faible.



Exemple avec la variable nombre qui a pour valeur 7, on obtiendra la valeur binaire 0111 :



Réalise le programme correspondant.

f) Vérifie le programme. Appelle le professeur quand le programme est valide.