

Fiche AD4	Nom :	Classe :	Date :
	ARDUINO UNO R3		
	Le radar de recul		

Activité 1 L'actionneur buzzer

Programme

- 1) A l'aide de la **fiche Arduino 4**, donner les commandes suivantes :
 - celles permettant d'allumer ou d'éteindre le buzzer :
.....
.....
 - celle permettant de mettre le programme en pause durant un temps t en s. (Librairie **time**)
.....



- 2) Compléter le programme suivant permettant d'allumer le buzzer durant 0,2 s.

Le buzzer est branché sur l'entrée **D4**.

Aide : On l'allume puis on met le programme en pause durant 0,2 s et on l'éteint.

```
from JeulinLib import*
from time import*
Connect("COM....")
SetBuzzerOff(4)      #Initialisation du buzzer
```

.....
.....
.....

Montage

- 1) Brancher sur l'ordinateur et initialiser l'**Arduino** (voir **Fiche Arduino 3**).
- 2) Brancher le **Grove – Actionneur Buzzer** sur l'entrée digitale **D4** de l'Arduino.
- 3) Lancer l'**interpréteur Python**.

Saisie et test

- 1) Clic sur **File** puis **New File**.
- 2) Dans la fenêtre qui s'ouvre, clic sur **File** puis **Save As...**, donner un nom et sauvegarder le fichier dans le dossier de destination.

Appel 1



- 3) Saisir le programme et le sauvegarder.
Lancer le programme : Clic sur **Run + Run Module**.

- 4) Modifier le programme avec une boucle bornée permettant de répéter 5 fois l'opération suivante :
Le buzzer est allumé durant 0,2 s puis éteint pendant 0,2 s.

```
for i in range(5):
```

.....
.....
.....
.....

Appel 2

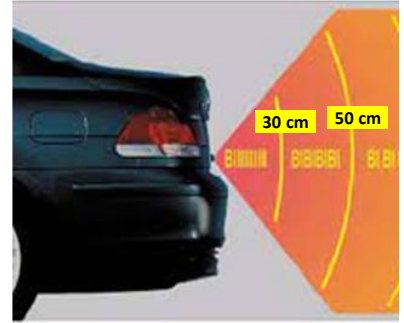


- 5) Saisir le programme et le sauvegarder.
Lancer le programme : Clic sur **Run + Run Module**.

Activité 2 Le radar de recul

Problème

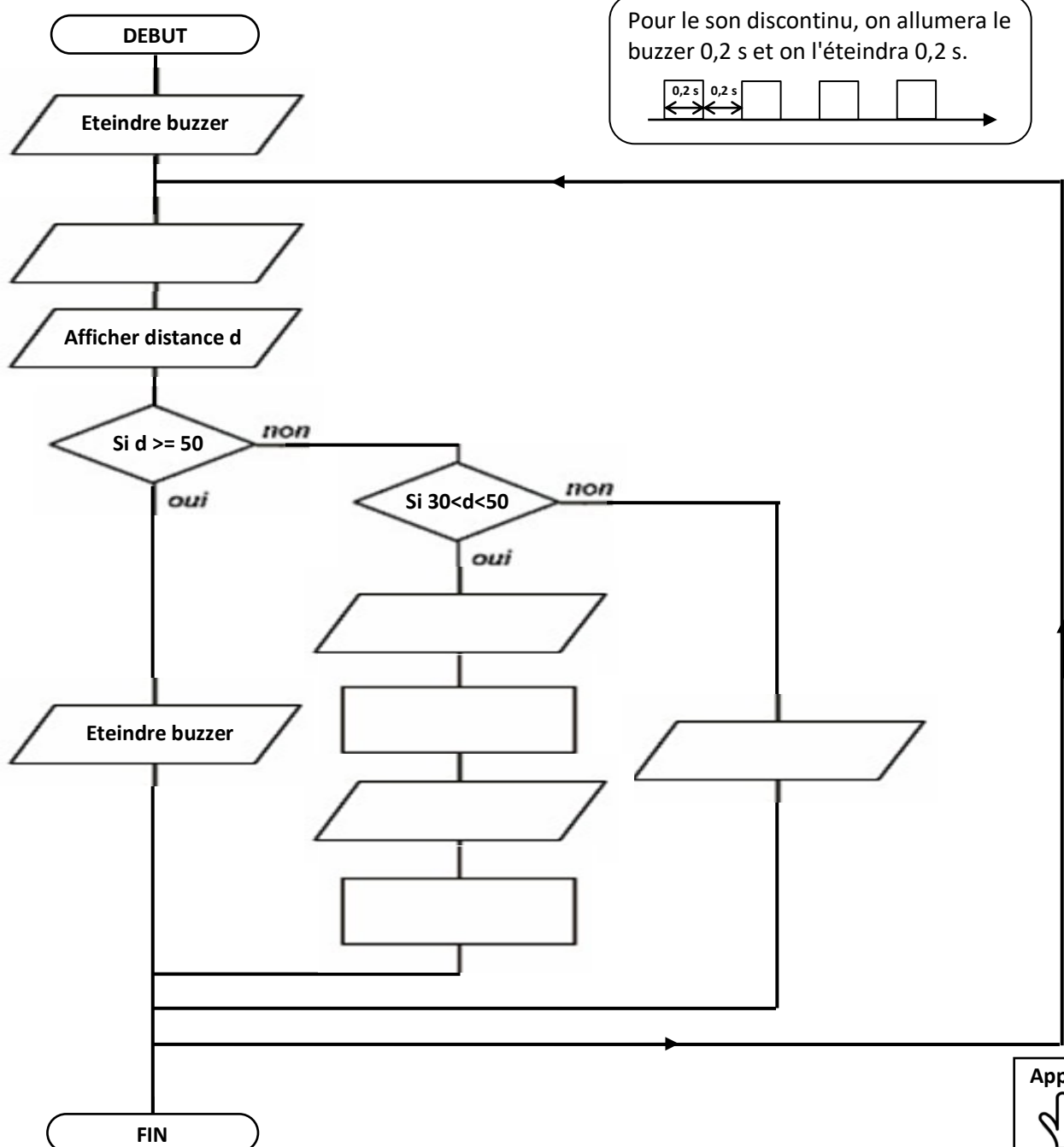
Le radar de recul qui équipe les automobiles consiste en un **capteur** (et un récepteur) **de distance à ultrasons** placé dans le pare-choc. Une fois mesurée la distance, un **buzzer** émet un son ou pas en fonction de la valeur de cette distance mesurée.



Si la distance est supérieure à 50 cm	Aucun son
Si la distance est comprise entre 30 et 50 cm	Son discontinu (bip-bip)
Si la distance est inférieure à 30 cm	Son en continu

Algorithme

Finir de compléter l'algorithme traduisant le problème.



Appel 3



Programme

- 1) A l'aide de la **fiche Arduino 4**, donner la commande du capteur US renvoyant la distance de l'obstacle en cm :



- 2) A partir de l'algorithme, écrire le programme correspondant.

Aides :

- Le capteur US est branché sur l'entrée **D8**.
- **"While true:"** est une boucle infinie. Pour la stopper : **Ctrl** + **C**
- On utilisera les conditions :
 - if**
 - elif:**
 - elif:**
- **"elif"** peut être traduit par **"sinon si"**
- Pour le son discontinu, on allumera le buzzer 0,2 s et on l'éteindra 0,2 s.

```
from JeulinLib import*
from time import*
```

```
Connect("COM....")
```

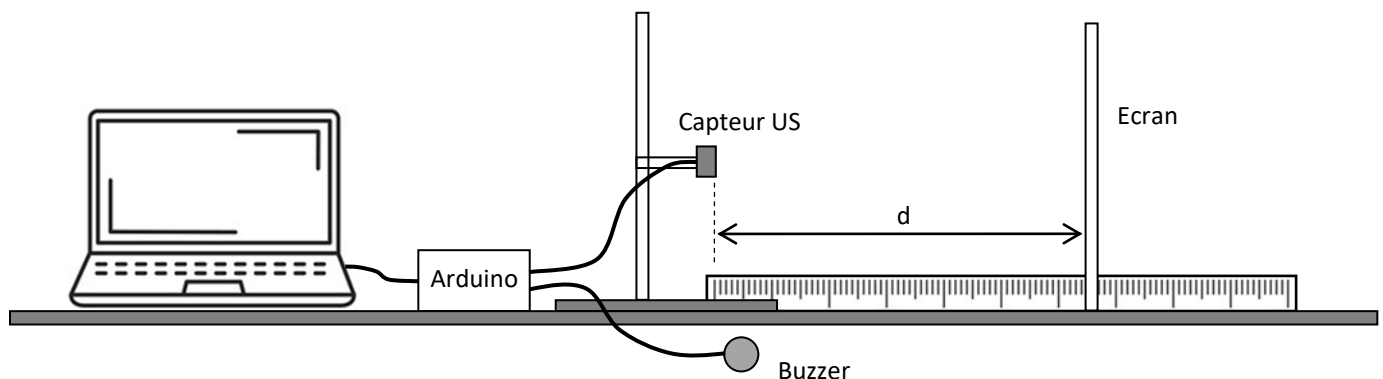
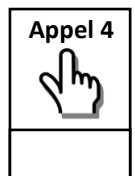
```
SetBuzzerOff(4)           #Initialisation du buzzer
```

```
while true:
```

[illegible]

Montage

- 1) Brancher sur l'ordinateur et initialiser l'**Arduino** (voir **Fiche Arduino 3**).
- 2) Brancher le **Grove – Capteur Buzzer** sur l'entrée digitale **D4** de l'Arduino.
- 3) Brancher le capteur **Grove – Capteur de distance à ultrasons** sur l'entrée digitale **D8** de l'Arduino.
- 4) Réaliser le montage ci-dessous.



Saisie

- 1) Lancer l'**interpréteur Python**.
- 2) Clic sur **File** puis **New File**.
- 3) Dans la fenêtre qui s'ouvre, clic sur **File** puis **Save As...**, donner un nom et sauvegarder le fichier dans le dossier de destination.
- 4) Saisir le programme et le sauvegarder.

Test

- 1) Positionner l'écran à une distance supérieure à 50 cm.
- 2) Lancer le programme : Clic sur **Run + Run Module**.
- 3) Rapprocher l'écran du capteur US afin de vérifier les distances.

Modification

Modifier le programme de telle manière :

Si la distance est supérieure à 70 cm	Aucun son
Si la distance est comprise entre 50 et 70 cm	Son discontinu long 
Si la distance est comprise entre 30 et 50 cm	Son discontinu court 
Si la distance est inférieure à 30 cm	Son en continu

