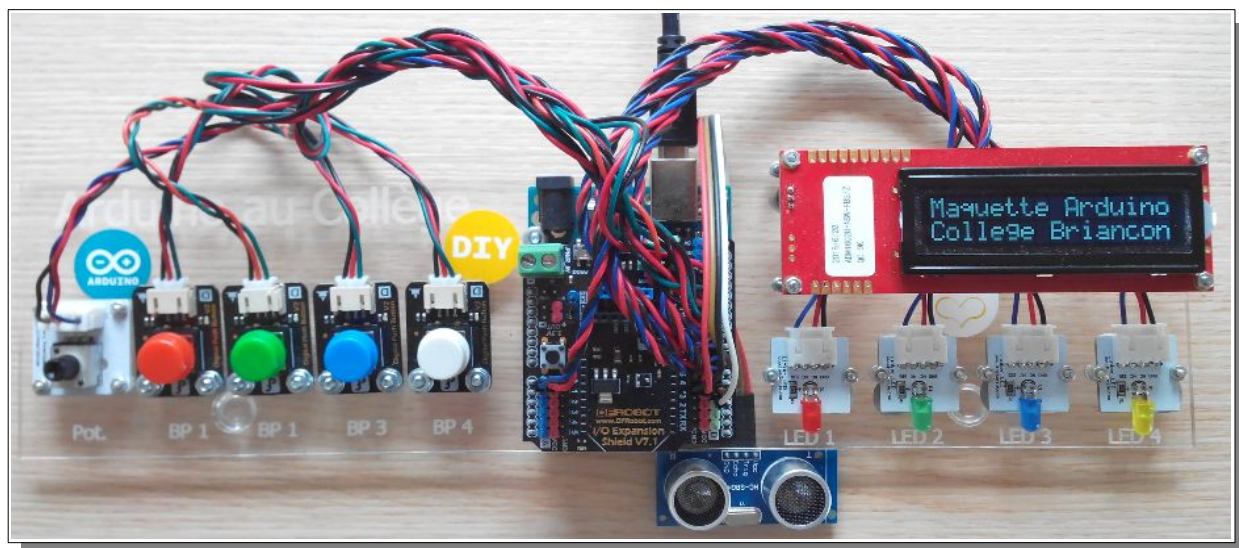


PREMIÈRE UTILISATION DE LA MAQUETTE

ARDUINO AU COLLÈGE

GESTION DES ENTRÉES LOGIQUES (BOUTONS POUSSOIRS) ET DES SORTIES LOGIQUES (LED).

1) LA MAQUETTE : « ARDUINO AU COLLÈGE »



Cette maquette a été conçue et réalisée par les professeurs de technologie des Collèges de Briançon. Elle a pour but de vous familiariser avec la programmation en langage structuré (le langage C dans notre cas) et de vous permettre de concevoir des systèmes évolués de manière simple et ludique.

La partie de gauche comporte les entrées :

- 1 potentiomètre (résistance variable qui permet de simuler des capteurs analogiques linéaires)
- 4 boutons poussoirs (interrupteurs monostables qui permettent de simuler des capteurs « Tout ou Rien »)

La partie centrale comporte la carte Arduino Uno. C'est une carte électronique basée sur un microcontrôleur « ATmega328 ». C'est la partie « intelligente » du système. C'est cette carte qui fait le lien entre les entrées et les sorties en respectant le programme contenu dans la mémoire du microcontrôleur. Elle dispose de :

- 14 broches numériques d'entrées/sorties
- 6 entrées analogiques
- une connexion USB
- etc.

La partie de droite comporte les sorties :

- 1 afficheur série 2x16 caractères
- 4 LED (témoins lumineux qui permettent de simuler des actionneurs « Tout ou Rien »)

Elle comporte aussi en périphérie :

- Un module ultrason (un émetteur + un récepteur qui permettent de mesurer une distance par mesure du temps mis par la salve ultrason à faire le trajet entre l'émetteur et le récepteur).

2) EXERCICE N°1 : IMPLANTER UN PROGRAMME DE TEST DANS LA MAQUETTE ET VÉRIFIER SON FONCTIONNEMENT.

L'ensemble des actions à faire avec la maquette se fait grâce au logiciel « Arduino » situé sur votre bureau.

1 **Lancez le logiciel « Arduino »**  à partir de votre bureau.

2 **Ouvrez le fichier :** « Sequenc1.ino » situé dans le dossier : « Technologie/3eme/Projet_1_RadarDeRecul/Sequenc1 ».

3 **Repérez-vous dans le programme :**

- La partie supérieure du programme contient des commentaires. Ces commentaires n'ont aucun rôle dans le programme. Ils sont juste présents pour donner des informations à la personne qui les lit. La zone de commentaire commence par les caractères /* et termine par les caractères */.

Remarque : **Les commentaires ont toujours une couleur grise**

```
/*
Séquence 1 : Première utilisation de la maquette: Gestion des entrées logiques (Boutons poussoir) et des sorties logiques (LED).
Utilisation de la fonction "if", de la fonction booléenne "ET" et des temporisations.

Le programme ci-dessous permet de lire l'état du bouton poussoir rouge (BP_Rouge).
L'état de ce bouton est envoyé sur la LED rouge (LED_Rouge).
=> Si le bouton poussoir rouge est appuyé alors la LED rouge s'allume
=> Si le bouton poussoir rouge est relâché alors la LED rouge s'éteint
*/
```

=> **Lisez les commentaires (directement dans le logiciel Arduino)**

- La partie inférieure du programme (sous le /*) contient le programme que doit réaliser la maquette :

```

//Bibliothèques utilisées
#include <SerialDisplay.h>
#include <Ultrasonic.h>
#include <MaquetteCollege.h>

/*-----*/
void setup()
{
  Initialisation();
}

void loop()
{
  if(digitalRead(BP_Rouge)==1)
  {
    digitalWrite(LED_Rouge, HIGH);
  }
  else
  {
    digitalWrite(LED_Rouge, LOW);
  }
}

```

Déclaration des bibliothèques utilisées dans le programme

Sous-programme « setup » :
Ce programme sera exécuté une seule fois au démarrage de la maquette. Il contient la demande d'initialisation de la carte.

Sous-programme « loop » :
Ce programme sera exécuté en boucle Indéfiniment. C'est le programme principal de la maquette.

=> Repérez les deux sous-programmes (directement dans le logiciel Arduino)

- Connectez la maquette sur un port USB de l'ordinateur. Au bout de quelques secondes la maquette est reconnue et le pilote activé.
- Paramétrez le port USB de programmation.

Sequence1 | Arduino 1.6.9
Fichier Édition Croquis Outils Aide

Formatage automatique Ctrl+T
Archiver le croquis
Réparer encodage & recharger
Moniteur série Ctrl+Maj+M
Traceur série Ctrl+Maj+L
Type de carte: "Arduino/Genuino Uno"
Port: "COM4 (Arduino/Genuino Uno)"
Get Board Info
Programmeur: "AVRISP mkII"
Graver la séquence d'initialisation

Ports série
COM3
COM4 (Arduino/Genuino Uno)

Sélectionnez le port COMx qui correspond au port utilisé par la carte Arduino.
On le détermine par la présence du texte (Arduino/Genuino Uno) à droite du n° de port COM.
Ici, c'est le COM4.
=> Choisissez le votre

- Faites le « téléversement ».

Sequence1 | Arduino 1.6.9
Fichier Édition Croquis Outils Aide

Téléverser

Cliquez sur l'icône en forme de flèche pour procéder à la compilation du programme puis à son téléchargement dans la maquette.

7 Analysez le fonctionnement du programme en appuyant sur le bouton rouge.

Que réalise ce programme ?

Validation du professeur :

3) EXERCICE 2 : ANALYSE DE LA STRUCTURE DU PROGRAMME POUR RÉALISER UNE PREMIÈRE MODIFICATION DU FONCTIONNEMENT.

3.1) CAHIER DES CHARGES DE L'EXERCICE :

Nous allons vous aider à modifier le programme pour que ça soit l'appui sur le bouton vert (*BP_Vert*) qui allume la LED rouge.

3.2) ANALYSE DU PROGRAMME :

Le programme à changer se situe dans la partie « loop » (la partie du programme qui boucle en permanence).

```
void loop()                                //Programme bouclé
{
  if(digitalRead(BP_Rouge)==1)             //Si la lecture de "BP_Rouge" donne "1" (on appuis sur le BP) :
  {
    digitalWrite(LED_Rouge, HIGH);         //Alors je place la "LED_Rouge" à l'état haut (HIGH)
  }
  else                                       //Sinon (si la lecture de "BP_Rouge" donne "0") :
  {
    digitalWrite(LED_Rouge, LOW);          //Alors je place la "LED_Rouge" à l'état bas (LOW)
  }
}
```

Le test de l'appui sur le bouton rouge se fait avec la ligne :

`if(digitalRead(BP_Rouge)==1)`

Pour changer de bouton poussoir, il faut donner le nom du bouton poussoir désiré.

3.3) ADAPTATION DU PROGRAMME AU NOUVEAU CAHIER DES CHARGES :

- 1 Changez le bouton poussoir à lire en remplaçant **BP_Rouge**, par **BP_Vert** dans le programme.
- 2 Faites le « téléversement » (faire comme le 6° de l'exercice précédent).
- 3 Vérifiez le fonctionnement du programme en appuyant sur les boutons poussoirs.
- 4 Si le programme répond au cahier des charges, Collez une copie d'écran de votre programme ci-dessous (ne garder seulement que la partie « loop »).

Validation du professeur :

4) EXERCICE 3 : MODIFICATION DU TÉMOIN DE SORTIE.

4.1) CAHIER DES CHARGES DE L'EXERCICE :

Nous allons vous aider à modifier le programme pour que l'appui sur le bouton vert allume la LED verte (LED_Verte).

4.2) ANALYSE DU PROGRAMME :

Le programme à changer se situe dans la partie « loop » (la partie du programme qui boucle en permanence).

L'allumage de la LED rouge se fait avec les lignes :

```
digitalWrite(LED_Rouge, HIGH);
```

et

```
digitalWrite(LED_Rouge, LOW);
```

Pour changer de LED à allumer, il faut donner le nom de la LED désirée.

4.3) ADAPTATION DU PROGRAMME AU NOUVEAU CAHIER DES CHARGES :

- 1 Changez la LED à allumer en remplaçant **LED_Rouge**, par **LED_Verte** dans le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, Collez une copie d'écran de votre programme ci-dessous (ne garder seulement que la partie « loop »).

Validation du professeur :

PRENONS DE L'AUTONOMIE

Jusqu'à présent, nous vous avons guidés pour faire les modifications. Dans cette deuxième partie vous devrez faire preuve d'analyse et d'initiatives pour faire les modifications demandées. Il n'est pas grave ni inquiétant de ne pas réussir du premier coup. Le résultat (et donc la note) vient en grande partie de la persévérance dans la recherche de solutions.

5) EXERCICE 4 : INVERSION DU FONCTIONNEMENT.**5.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appuie sur le bouton vert éteigne la LED verte.

5.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, **Collez une copie d'écran de votre programme ci-dessous** (ne garder seulement que la partie « loop »).

Validation du professeur :

6) EXERCICE 5 :**6.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appui sur le bouton vert éteigne la LED verte **et** allume la LED rouge.

6.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, **Collez une copie d'écran de votre programme ci-dessous** (ne garder seulement que la partie « loop »).

Validation du professeur :

7) EXERCICE 6 :**7.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appui sur le bouton rouge allume la LED rouge, et que l'appui sur le bouton vert allume la LED verte.

7.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, **Collez une copie d'écran de votre programme ci-dessous** (ne garder seulement que la partie « loop »).

Validation du professeur :

8) EXERCICE 7 :**8.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appui sur **chaque bouton** allume la LED correspondante.

8.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, **Collez une copie d'écran de votre programme ci-dessous** (ne garder seulement que la partie « loop »).

Validation du professeur :

9) EXERCICE 8 :**9.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour qu'il soit nécessaire d'appuyer sur le bouton rouge ET sur le bouton blanc pour que la LED rouge soit allumée

9.2) ANALYSE ET MODIFICATION DU PROGRAMME

Remarque importante: Vous pouvez chercher sur internet pour trouver la syntaxe de l'opérateur logique « ET » en langage C, et notamment sur le site suivant :

<https://arduinogetstarted.com/fr/reference/arduino-logical-and>

En résumé la syntaxe d'un ET logique est : ((condition1) && (condition2))

avec « condition1 » qui pourrait être, par exemple, « digitalRead(BP_Rouge)==1 »

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, Collez une copie d'écran de votre programme ci-dessous (ne garder seulement que la partie « loop »).

Validation du professeur :

10) EXERCICE 9 :**10.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appui sur le bouton rouge fasse clignoter la LED rouge avec une fréquence de 1 Hz (500 ms éteinte puis 500 ms allumée).

10.2) ANALYSE ET MODIFICATION DU PROGRAMME

Remarque importante: Pour générer un retard (ou une attente) fixe sur un programme, il est possible d'utiliser la fonction « delay() ».

Exemple : La ligne :

```
delay(1000);
```

génère une attente de 1000 milli-secondes (soit 1 seconde).

- 1 Analysez puis modifiez le programme.**
- 2 Faites le « téléversement ».**
- 3 Vérifiez le fonctionnement du programme.**
- 4 Si le programme répond au cahier des charges, Collez une copie d'écran de votre programme ci-dessous (ne garder seulement que la partie « loop »).**

Validation du professeur :

POUR ALLER PLUS LOIN...

Dans cette troisième partie vos capacités d'analyses et de concentration devront être fortes. Il est donc normal de passer davantage de temps pour réussir une question. Il faut rester focalisé sur ce qui est demandé, bien analyser la situation, faire des hypothèses et des essais, puis essayer de comprendre ce qui a fonctionné et ce qui n'a pas répondu à votre attente pour pouvoir recommencer et, à force de volonté, réussir. Il n'est pas grave ni inquiétant de ne pas réussir du premier coup. Le résultat (et donc la note) vient en grande partie de la persévérance dans la recherche de solutions.

11) EXERCICE 10 : RÉALISATION D'UN CHENILLARD.**11.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que l'appui sur le bouton rouge fasse un chenillard avec toutes les LED (Durée d'allumage par LED : 200 ms)

Qu'est-ce qu'un chenillard : <https://wiki.centrale-marseille.fr/fablab/start:projet:arduino:formation>

11.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.**
- 2 Faites le « téléversement ».**
- 3 Vérifiez le fonctionnement du programme.**
- 4 Si le programme répond au cahier des charges, Collez une copie d'écran de votre programme ci-dessous (ne garder seulement que la partie « loop »).**

Validation du professeur :

12) EXERCICE 11 :**12.1) CAHIER DES CHARGES DE L'EXERCICE :**

Modifier le programme pour que le **chenillard** soit de vitesse différente en fonction du bouton appuyé :

- **Si BP_Rouge appuyé:** La totalité du chenillard à une durée d'allumage par LED de 1 s
- **Si BP_Vert appuyé:** La totalité du chenillard à une durée d'allumage par LED de 500 ms
- **Si BP_Bleu appuyé:** La totalité du chenillard à une durée d'allumage par LED de 200 ms
- **Si BP_Blanc appuyé:** La totalité du chenillard à une durée d'allumage par LED de 100 ms

12.2) ANALYSE ET MODIFICATION DU PROGRAMME

- 1 Analysez puis modifiez le programme.
- 2 Faites le « téléversement ».
- 3 Vérifiez le fonctionnement du programme.
- 4 Si le programme répond au cahier des charges, **Collez une copie d'écran de votre programme ci-dessous** (ne garder seulement que la partie « loop »).

Validation du professeur :

13) AJOUT DE LA SÉQUENCE 1 À VOTRE CLASSEUR NUMÉRIQUE

A la fin de chaque séquence, vous devrez intégrer votre fichier de la séquence finie (le fichier « activites.odt » dans lequel vous avez travaillé) dans votre classeur numérique.

Pour cela vous devez :

- Générer un fichier PDF à partir de votre fichier traitement de texte.
- Intégrer la séquence 1 (que vous venez de générer) à votre classeur numérique déjà existant.

Pour vous aider, vous avez le fichier « Classeur_numerique.pdf » présent dans les ressources de la séquence 0 du projet 1 (ou au début de votre classeur numérique).