



Interface Homme Machine

1 .Présentation

Qu'appelle-t-on une interface Homme Machine ?

Il s'agit d'une interface permettant à un utilisateur d'interagir avec une machine.

Les IHM ont de nombreuses utilisations, notamment dans le monde industriel :

Suivre en temps réel les productions, contrôler les températures, commander tout type de robots.

La souris, le clavier, l'écran tactile d'un téléphone, une télécommande sont différents types d'interface.

Par machine, on peut entendre ordinateur mais aussi robot, que l'on trouve à foison dans le monde industriel. Tous ces objets ont un point commun, ils possèdent un **système informatique embarqué : Des capteurs, des actionneurs et une carte programmable .**

- Les actionneurs : Ce sont des composants qui agissent sur un système pour en modifier le comportement. Les actionneurs transforment les informations reçues de la carte programmable : Trier des pièces, peindre une portière, allumer une lampe ...
- Les capteurs : Ils permettent à la machine de connaître son environnement : Proximité d'un mur, capteur de température... Les capteurs envoient des données exploitables à la carte programmable, qu'elle exploite par la suite.

Pour interagir avec une machine, on peut la programmer par exemple. Les cartes programmables sont munies de micro contrôleurs capables d'analyser et d'exécuter les programmes qu'on leur transmet.

Nous allons par la suite travailler avec une carte Arduino, que nous allons présenter ainsi que, sommairement, le langage de programmation adéquat.

Pour éviter tout souci matériel, nous allons travailler avec le simulateur Tinkercad et présenter aussi un exemple sans programmation, où l'actionneur sera un bouton.

<https://www.tinkercad.com/>

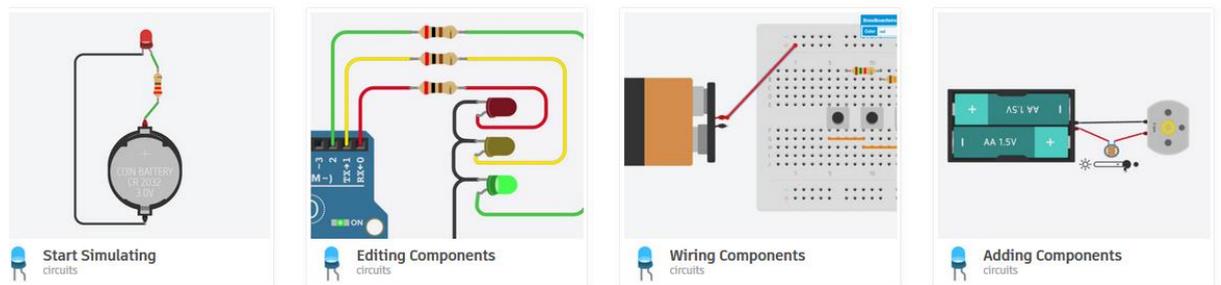
2 .Simulation avec Tinkercad

Nous allons maintenant découvrir le simulateur qui va nous accompagner sur ce chapitre.

Il y a une section « Apprendre » qui va nous permettre de nous familiariser avec certains composants que nous utiliserons plus tard.

Circuits- Kits de démarrage Leçons Projets

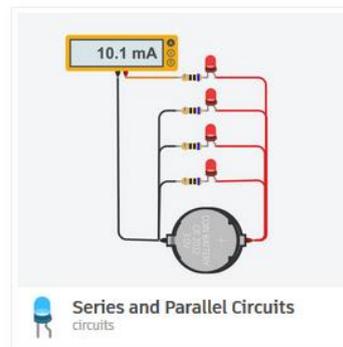
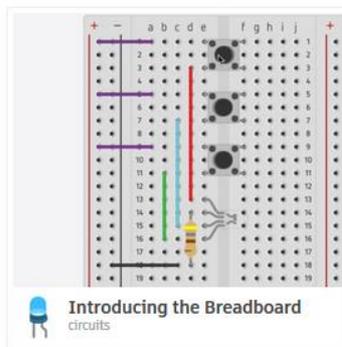
Les guides de mise en route contiennent des leçons rapides et simples pour apprendre les notions de base sur les circuits Tinkercad.



The image shows four tutorial cards from the Tinkercad 'Circuits' section. Each card features a small circuit diagram and a title with a blue robot icon. The titles are: 'Start Simulating circuits', 'Editing Components circuits', 'Wiring Components circuits', and 'Adding Components circuits'. The diagrams illustrate basic circuit connections, including a battery, resistors, LEDs, and a breadboard.

A faire préalablement : Tester ces quatre tutoriels qui vont vous permettre de comprendre le fonctionnement du simulateur, d'ajouter des composants, d'éditer les composants.

Vous pouvez jeter un œil aux tutoriels plus techniques orientés « cours de physique » avec notamment le rappel de la loi d'Ohm. Ces tutoriels sont accessibles dans la rubrique circuits, leçons.

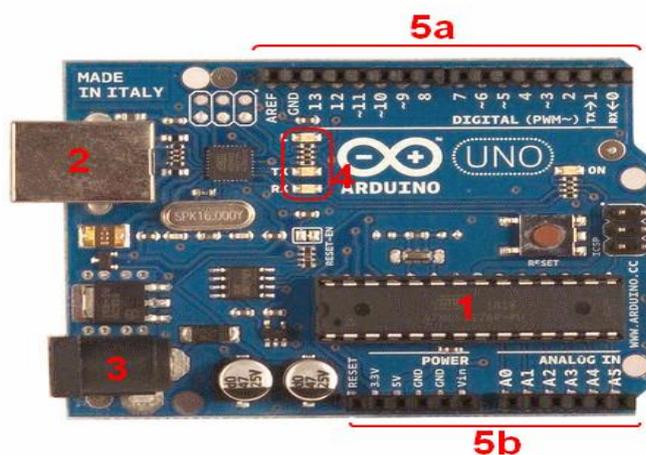


Maintenant que l'on maîtrise les outils de base, voyons comment fonctionne une carte Arduino et comment on peut l'utiliser avec nos composants.

3. La carte Arduino

On va ici présenter la carte Arduino Uno et ses principales composantes.

1. C'est le micro contrôleur, le cerveau de la carte. Il reçoit, analyse, stocke et exécute le programme qu'on lui envoie.
2. 3. Ce sont les alimentations pour la carte. Soit par le port USB (2) soit par une alimentation externe (3). La carte fonctionne en 5 V et possède des régulateurs pour adapter la tension reçue.
4. Ce sont des LED de contrôle. Celle en haut permet de tester le matériel. Elle clignote lors de la mise sous tension de la carte. Les deux autres clignotent lors du téléchargement de programmes.
5. Ce sont les connectiques de la carte, c'est à dire, l'endroit où l'on pourra connecter les composants qui nous intéressent pour la réalisation de nos projets.



4. TP

TP 1 : On va, humblement faire clignoter une lampe de deux façons .

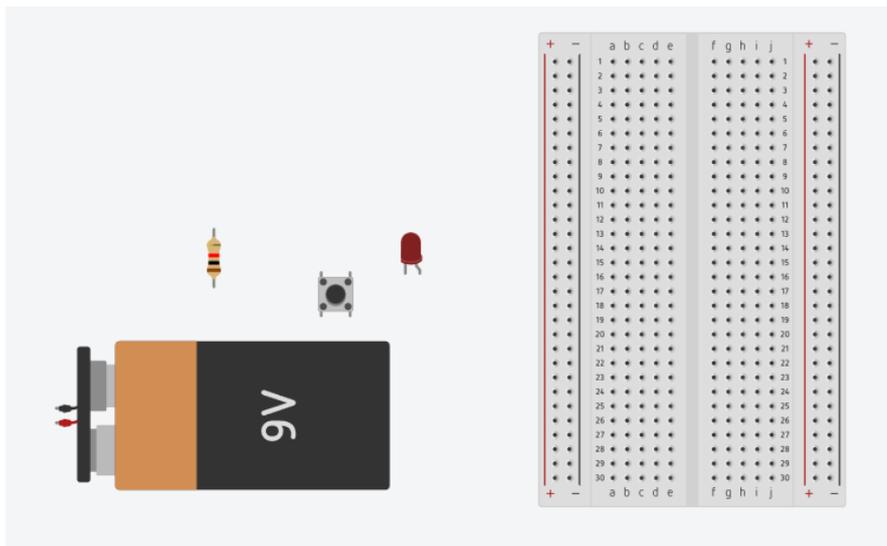
1. A l'aide du breadbord ... et d'un bouton poussoir.

2. A l'aide de la carte Arduino. Dans ce cas, il nous faudra écrire un petit code (mais on pourra garder les mains dans les poches, on n'a rien sans rien!)

TP1 breadbord

Voici les composants dont vous avez besoin : Si vous avez regardé les tutos, c'est la ballade assurée, sinon, vous auriez dû...

Pour ceux qui sont les plus avancés, installez deux voire trois leds de couleurs différentes.



TP 1 Arduino.

Ici, on a juste besoin de la carte Arduino, puisque la led qui teste le bon fonctionnement de la carte est reliée à la connectique 13. A nous de programmer la carte pour faire clignoter cette carte.

Programmation avec Arduino .

Pour tout programme Arduino, il y a un code minimal qui se présente comme suit :

```
1 // fonction d'initialisation de la carte
2 void setup()
3 {
4     // contenu de l'initialisation
5 }
6
7 // fonction principale, elle se répète (s'exécute) à l'infini
8 void loop()
9 {
10     // contenu de votre programme
11 }
```

Ici, on a besoin d'initialiser la led qui va clignoter.

Dans la fonction principale, il faudra l'allumer, donner le temps où elle reste allumer, puis l'éteindre et donner le temps où elle reste éteinte.

Voilà, super simple . Mais comment ça se dit en Arduino ?

Dans la partie setup, on va indiquer que l'on a une sortie (allumage de la lampe) sur la broche 13 .

Sortie se dit OUTPUT (entrée : INPUT). La fonction à utiliser est pinMode.

PinMode(n°de broche, Entre/Sortie)

Dans la partie loop, on veut allumer la lampe et donner le délai :

Pour allumer ou éteindre : **digitalWrite** avec en paramètre le N° de broche et la valeur HIGH ou LOW. (A vous de deviner qui fait quoi ...)

La fonction **delay** donne le temps d'exécution avec en paramètre le temps en millisecondes.

Avec tout ça, cela devrait fonctionner.

Remarque : Chaque instruction se termine par un point virgule

TP personnel : Réaliser un montage utilisant une carte Arduino, une breadbord et une led pour obtenir une led qui clignote. On utilisera la broche n°2 et une alimentation de 5V depuis la carte Arduino. Il vous faudra peut être ajouter une résistance.

TP 2 : Réaliser une simulation de synchronisation de deux feux de signalisation.

On créera un programme pour une carte Arduino reliée à une Breadboard configurée pour réaliser la simulation.

Matériel nécessaire :

- Une carte Arduino.
- 6 Leds (2de chaque couleur du feu)
- 6 résistances
- Et le câblage

5. Pour aller plus loin

Pour réaliser des montages autrement plus complexes, il va falloir s'attaquer à du codage plus poussé. Comme dans la majorité des langages, le langage Arduino permet l'utilisation de conditions, de boucles, fonctions....

Lorsque l'on travaille avec une vraie carte, on doit programmer via une interface logicielle que l'on peut télécharger sur le site officiel d'Arduino

Download the Arduino IDE



<https://www.arduino.cc/en/Main/Software>

On peut ensuite via ce logiciel créer nos programmes .

```
sketch_aug06a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Une fois créé, on peut copier coller le programme dans le simulateur. Si l'on travaille avec une vraie carte, il suffit de le téléverser, en ayant bien précisé le matériel utilisé, le port de connexion...

Enfin pour programmer et connaître les mots clés du langage Arduino, dans la rubrique aide, tapez sur référence et miracle (si vous êtes connectés ...) vous arrivez sur cette page où chaque instruction est détaillée.

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- `setup()`
- `loop()`

Control Structures

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`

Variables

Constants

- `HIGH` | `LOW`
- `INPUT` | `OUTPUT` | `INPUT_PULLUP`
- `LED_BUILTIN`
- `true` | `false`
- integer constants
- floating point constants

Data Types

Functions

Digital I/O

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`

Analog I/O

- `analogReference()`
- `analogRead()`
- `analogWrite()` - *PWM*

Remarque : Le logiciel propose pleins d'exemples de code utilisables directement.

6.TP avancé

On va réaliser un capteur de température.

Matériel à utiliser :

- Breadbord
- Carte Arduino
- Capteur de températures
- Leds
- Resistances

Effectuer le montage sur le simulateur.

Créez le code qui permet suivant les températures d'allumer telle ou telle led.

Source

<https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/>