

Arduino

Vincent DEJONGHE
Amélie LAPORTE
Adrien MENDES SANTOS
Rodolphe TWOREK

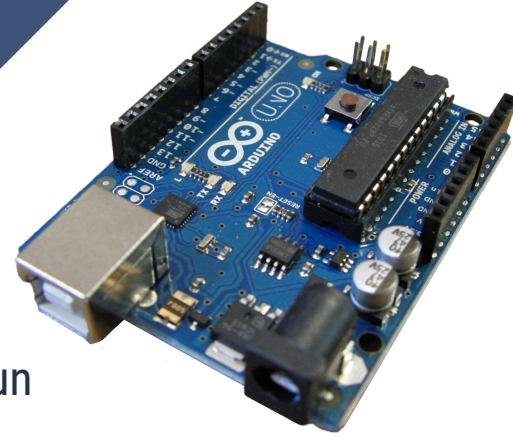
université
de **BORDEAUX**

Suivez le cours sur:

<https://anthagonas.github.io/arduino/>



Qu'est ce que Arduino?



Un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur.



Une petite unité de calcul accompagné de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement.



Les atouts

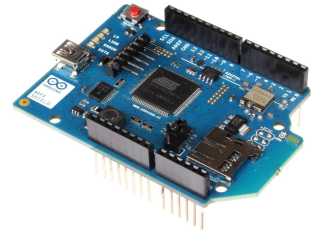
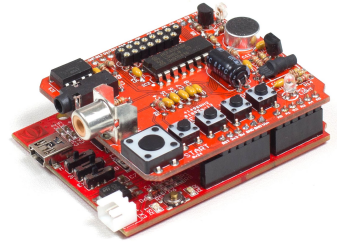
- Shields:

Une carte que l'on ajoute directement à son Arduino qui est livrée avec une bibliothèque permettant de le contrôler.



- IDE Arduino.

- Langage de programmation.





Programmer avec Arduino



Setup: Toutes les opérations nécessaires à la configuration de la carte.

```
void setup(){  
}  
void loop(){  
}
```

Loop: Exécute en boucle après l'exécution de la fonction setup.



Les variables

Nom	Contenu	Plage de valeurs
char	Entier ou caractère	(0 -> 255) -128 -> 127
int	Entier	(0 -> 65 535) -32 768 -> 32 767
long	Entier	(0 -> 4 294 697 295) -2 147 483 648 -> 2 147 483 647
Float / double	Nombre à virgule Flottante	-3,4028235 E+38 -> 3,4028235E+38
string	Chaîne de caractères (Objet)	Aucune
Booléen	Booléen	true/false



Les opérateurs

=	affectation
==	comparaison
!=	différence
<, >	Inférieur, supérieur
<=	Inférieur ou égal
>=	Supérieur ou égal
&&	et
	ou
!	non

Mathématique

+	plus
-	moins
*	multiplication
/	division
%	modulo



Les boucles

If:

```
if(<valeur booléenne>){  
  <instruction>;  
}else{  
  <instruction>;  
}
```

Switch:

```
Switch(<variable>){  
  Case <valeur>:  
    <instruction>;  
  Break;  
  default:  
    <instruction>;  
}
```



Les boucles

While:

```
while(<valeur booléenne>)  
{  
  <instruction>;  
}
```

For:

```
for(<initialisation>;<valeur  
booléenne>;<évolution>)  
{  
  <instruction>;  
}
```




Fonctions , structures et classes

Fonction:

```
<type> <nom>(<paramètre>)  
{  
<instruction>;  
}
```

Structure :

```
struct <nom> {  
<type> <nom du  
champ>;  
}
```

Classe:

```
class  
<nom> {  
    public:  
        <attributs et champ  
        publics>;  
    private:  
        <attributs et champ  
        privés>;  
    protected:  
        <attribut et champ privés>;  
}
```



Fonctions de bases

- Fonctions de base pour l'utilisation d'Arduino.
- Mesure/Envoi de tension sur les broches.
- Définir en amont les broches (pinMode).



```
void setup() {  
  pinMode(1,OUTPUT); //  
  Broche 1 en sortie  
  pinMode(2,INPUT ) ; //  
  Broche 2 en entrée  
}
```



Entrées / Sorties

Différentes fonctions pour utiliser les broches:

- **digitalRead(pin):** Broche en entrée. Mesure une valeur numérique sur une de broche.
- **digitalWrite(pin,value):** Broche en sortie. Écrit une valeur numérique sur une broche. Le paramètre value doit être égal à HIGH (état 1 soit 5V) ou LOW (état 0 soit 0V).



Entrées / Sorties

Toutes les broches sont capables d'écrire et de lire des valeurs numériques (c'est-à-dire des 0 (0V) ou des 1 (5V)).

Il existe un type de broche, les broches Pulse Width Modulation(PWM), permettant de reproduire un effet analogique.



La gestion du temps

Fonctions permettant de gérer le temps.

- **delay et delayMicroseconds :**

Pause. Insère une pause suivant le paramètre passé, bloque le microcontrôleur, on ne peut alors plus effectuer aucune action.

- **millis et micros :**

Mesure du temps. Permet d'incrémenter un intervalle pour une action donné. Ces fonctions incrémente une variable.



Les interruptions

Événement externe (appui sur un bouton, données d'un capteur, etc.) pour effectuer une action.

Utilisation des fonctions d'interruptions :

- **attachInterrupt et detachInterrupt**

Ce mécanisme interrompt le code exécuté, il est prioritaire par rapport au reste du code.



Les évènements

4 types d'évènements :

HIGHT et LOW: Renvoie l'état de voltage de la broche.

RISING: Lorsque la broche passe de l'état 0 (0V) à l'état 1 (5V) (front montant).

FALLING: Lorsque la broche passe de l'état 1 (5V) à l'état 0 (0V) (front descendant).

CHANGE: Lorsque la broche change d'état (front montant et front descendant).



Les interruptions - Exemple

```
volatile boolean etat=false;

void appuiBouton(){
    etat=!etat; //Changement d'état
}

void setup(){
    pinMode(2,INPUT); //Broche 2 en entrée

    attachInterrupt(0,appuiBouton,RISING); // On attache à
l'interruption 0 (broche 2) la fonction appuiBouton sur un front montant
}
```