Présentation des plateformes Arduino

AZAYTECH 2023

Plan du cours

1. Arduino, pour quoi faire ?
2. Alternatives
3. La famille Arduino
4. UNO ou NANO : les entrailles
5. Autonome ou communiquant
6. Se documenter
7. En pratique : programmer, l'IDE point de départ
8. En pratique : C++ le langage
9. En pratique : les bibliothèques
10. Mise en oeuvre : commander un servomoteur

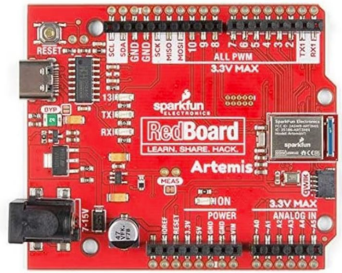
1 - Arduino, pour quoi faire ?

Plateforme de **prototypage** open-source qui permet aux utilisateurs de créer des **objets électroniques interactifs** à partir de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due).

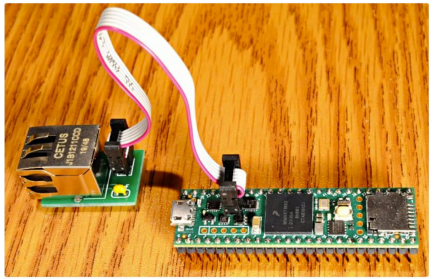
2 - Alternatives

BB52 Explorer Kit

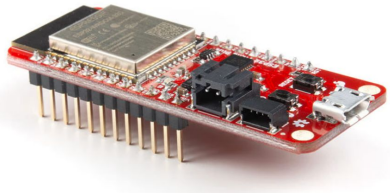
SparkFun RedBoard Artemis



Teensy 4.1

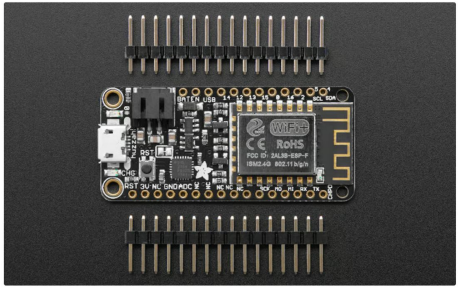


SparkFun Thing Plus



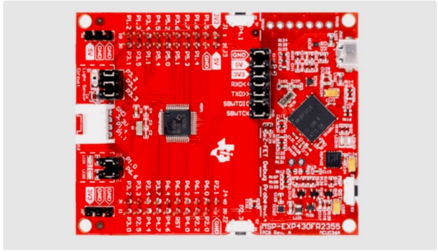
Esp32 DevKitC-VIE

Adafruit Feather Huzzah



Esp8266 DevKitC

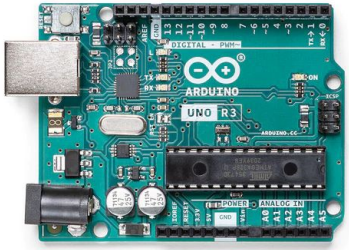
Launchpad MSP430



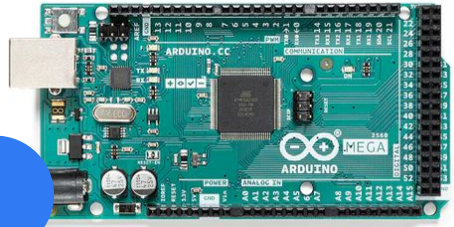
BBC Micro:bit V2

3 - La famille Arduino

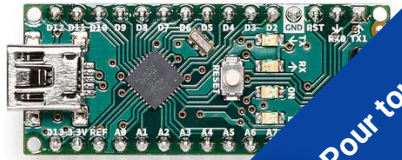
UNO

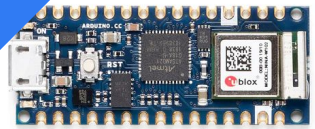


MEGA



NANO





1 Pour tout le monde

2 Pour les professionnels

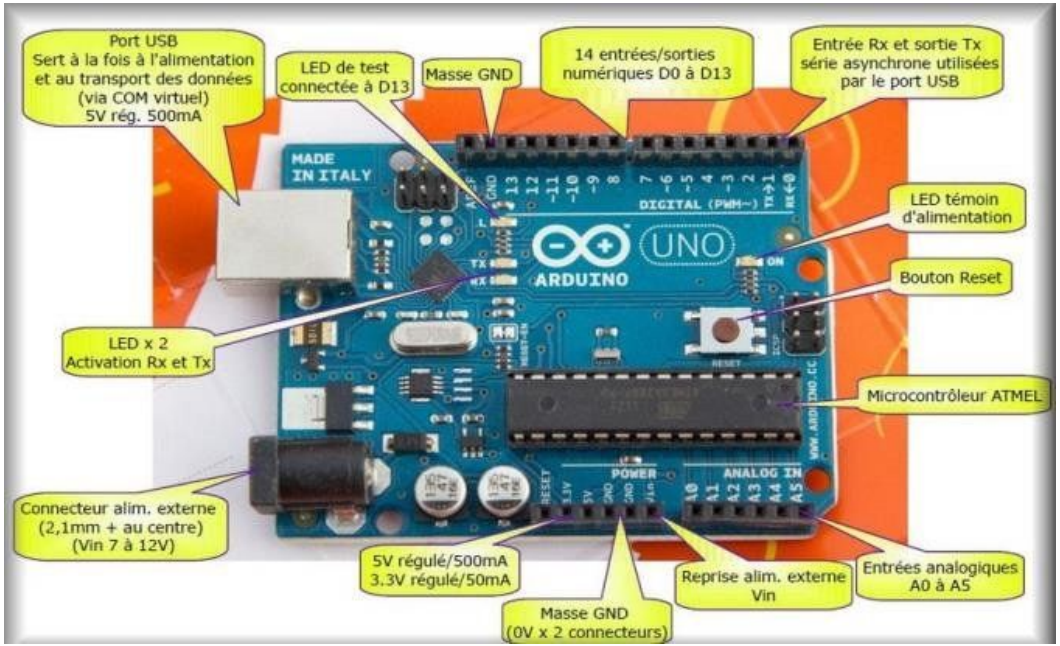
3 Pour l'éducation

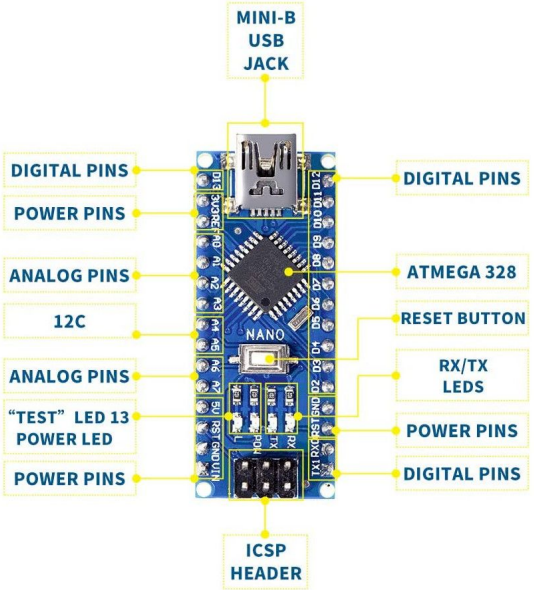
4 Pour l'IOT

Portenta H7

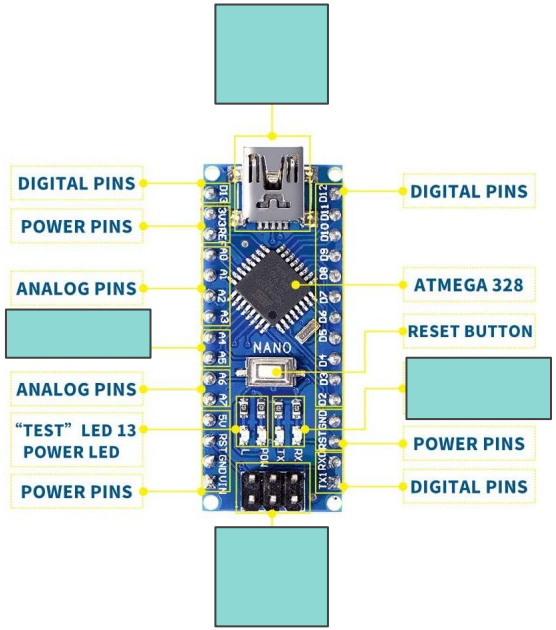
Shields

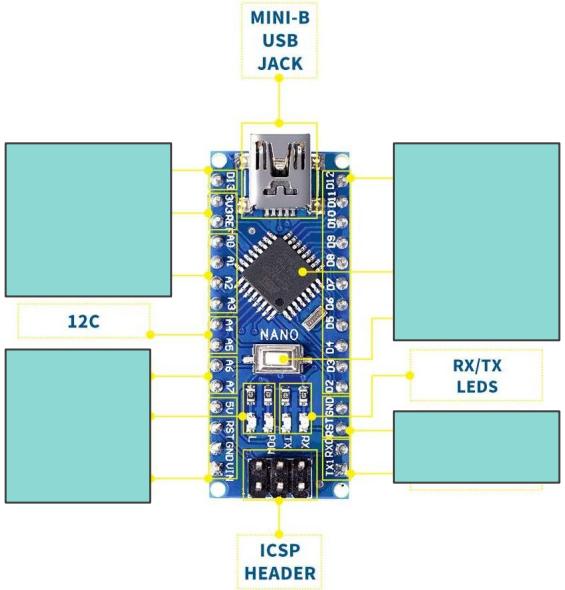
4 - UNO ou NANO : les entrailles





5 - Autonome ou communiquant



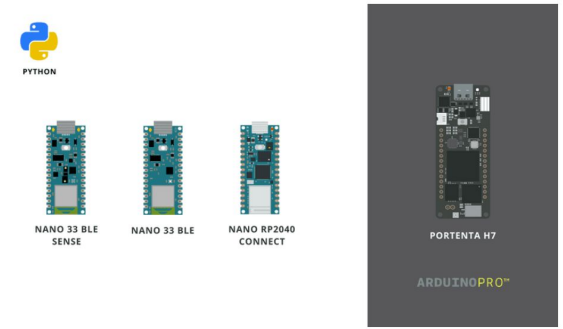


6 - Se documenter

Arduino : https://docs.arduino.cc/

PROGRAMMER :

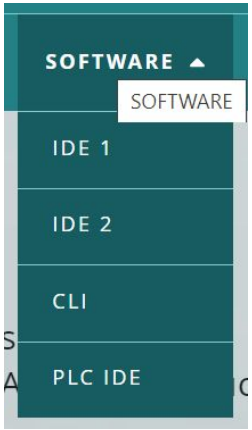
- Fonctions
  - calcul +/- complexe à appeler au besoin
- Variables
  - données à manipuler
- Structures
  - itérer, tester, conditions,...
- Librairies
  - adaptée à un usage précis
  - méthodes et attributs
  - modèle objet
- Exemples
- PYTHON



Arduino boards officially supporting MicroPython.

7 - En pratique : programmer, l'IDE point de départ

IDE : Un environnement de développement intégré (IDE) est une application logicielle qui aide les programmeurs à développer efficacement le code logiciel.

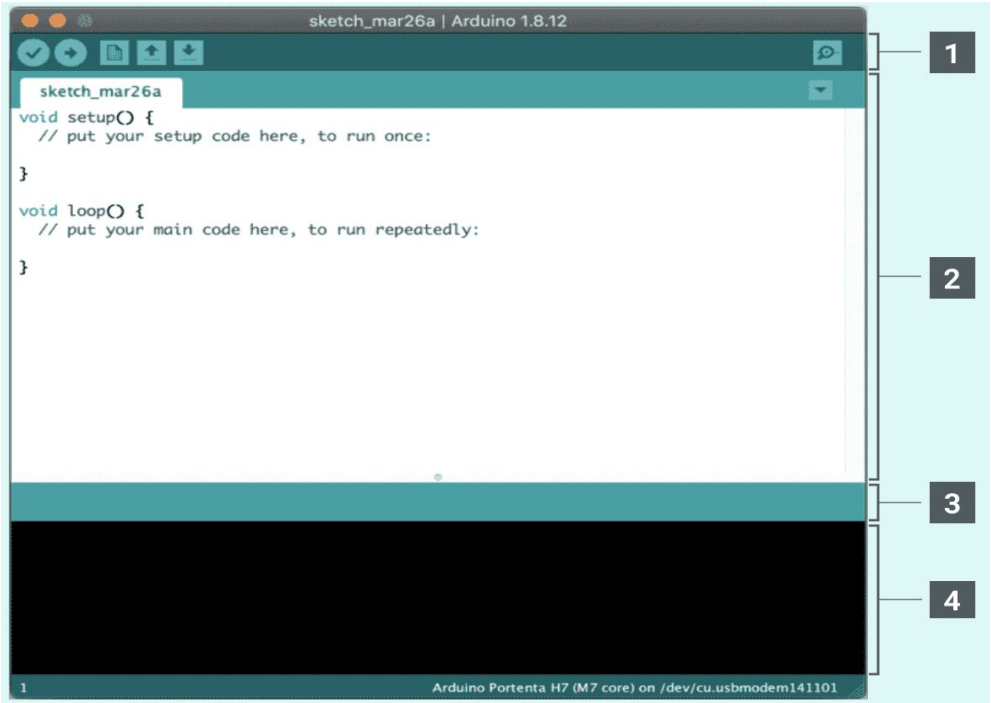


Pratique :

- coloration syntaxique
  - fonctions et mots réservés
- organisation du code et des fichiers
  - répartition du code en fichiers
- analyse avant compilation
  - erreurs courantes, ou pas
- pilotage du transfert
  - liste des cartes, pilotes

\+ le web editor, pour travailler en ligne

7.1 - En pratique : programmer, l'IDE point de départ

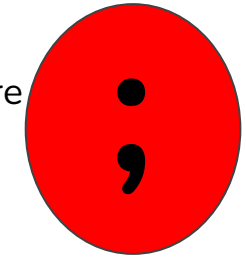


1. Une **barre d'outils** avec des boutons pour les fonctions courantes et une série de menus. Les boutons de la barre d'outils vous permettent de vérifier et de télécharger des programmes, de créer, d'ouvrir et d'enregistrer des croquis et d'ouvrir le moniteur série.
2. L' **éditeur de texte** pour écrire votre code.
3. La **zone de message**, donne des commentaires lors de l'enregistrement et de l'exportation et affiche également les erreurs.
4. La **console de texte** affiche la sortie texte du logiciel Arduino (IDE), y compris les messages d'erreur complets et d'autres informations.

Le coin inférieur droit de la fenêtre affiche la carte et le port série configurés.

7.1 - En pratique : C++ le langage

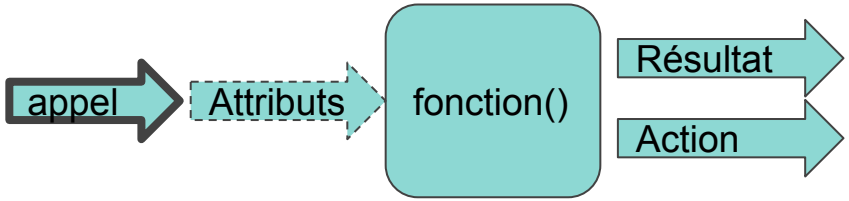
Votre pire ennemi



Les fonctions structurantes :
setup() : une seule fois à l'initialisation
loop() : exécutée en boucle infinie

Le sketch (croquis) :

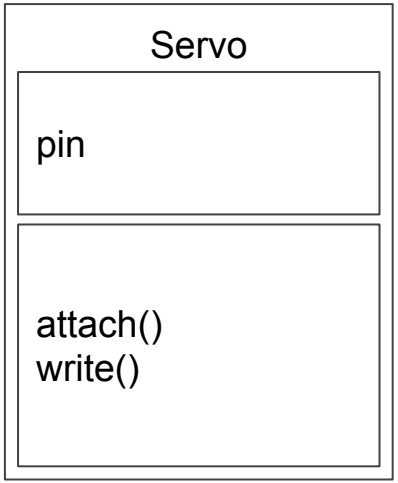
- déclarations et inclusions = variables, constantes, librairies, fichiers
- setup()
- loop()
- fonctions = bloc de code, réutilisables à l'infini
  - utilisent des attributs (option)
  - renvoient un résultat (option)



9 - En pratique : les bibliothèques

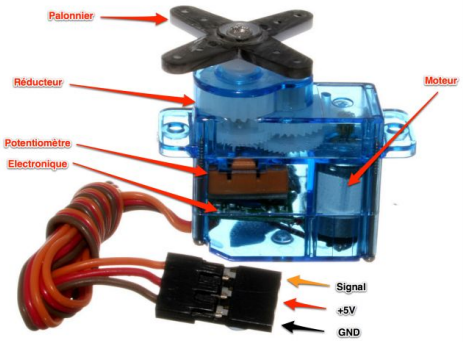
Bibliothèques (library)

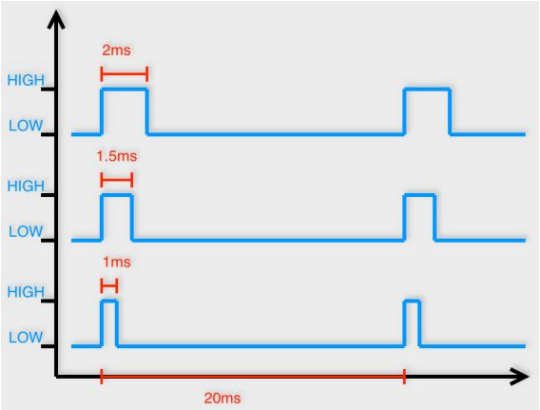
- travail fait par quelqu'un d'autre : génial !
- usage précis et bien étudié : concurrence
- modèle objet :
  - instancier un objet
  - attributs
  - méthodes

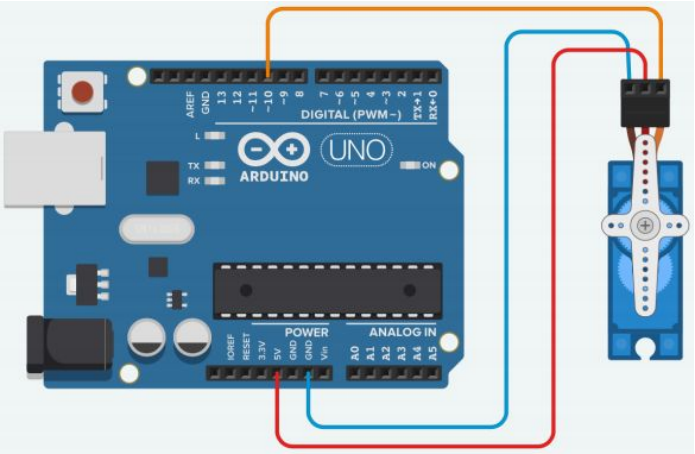



```
1  ##include <Servo.h>  // inclusion de la librairie Servo
2
3  Servo myservo;       // déclaration d'une variable de type Servo
4
5  void setup()
6  {
7    myservo.attach(9); // attache la variable Servo à la broche 9
8  }
9
10 void loop() {}
```

10 - Mise en oeuvre : commander un servomoteur







Servo.h

TP

1. installer l'IDE 2
2. câbler le servo
3. programmer :
    a. mise à zéro
    b. balayer de 0 à 180°
    c. revenir à 0