

Module 01 : Introduction à la programmation d'objets connectés

Applications mobiles et objets connectés 420-W48-SF

Objectifs

- Présentation de quelques outils
- Introduction aux microcontrôleurs
- Architecture du microcontrôleur Arduino
- Installation de l'environnement de programmation Arduino IDE
- Structure d'un programme pour Arduino
- Installation et exécution d'un programme dans un microcontrôleur

Outils individuels prêtés

- Tournevis d'horloger
- Pinces pour différentes utilités
- « Troisième main »: support avec loupe pour souder
- Loupe éclairante



Outils communs

- Multimètre
- Oscilloscope
- Bloc d'alimentation variable
- Microscope de table
- Fusil à colle chauffée
- Et Plus



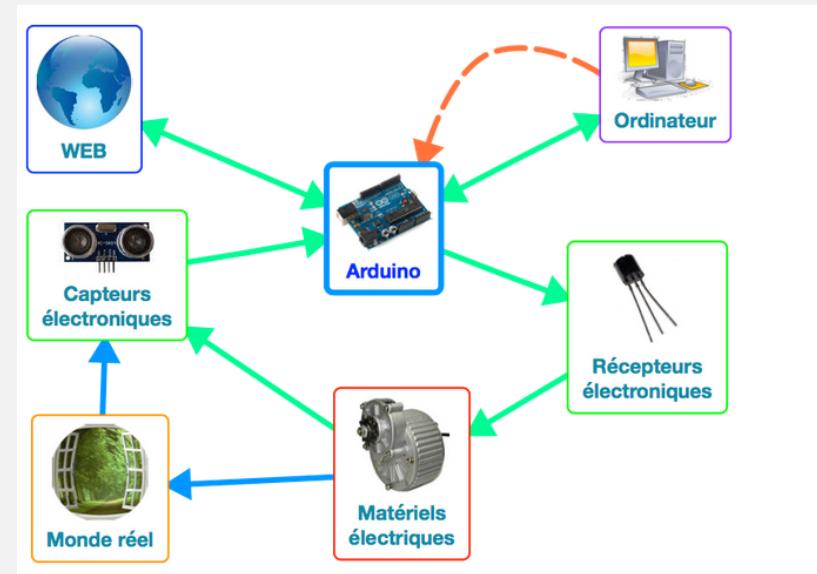
Station de soudure

- Station de soudure
- Petite bobine de fibre à soudure
- Protection de la pointe (panne) de soudure
- Tresse à dessouder
- Ventilateur de protection



Introduction aux microcontrôleurs

- Petits circuits électroniques pouvant être programmés pour interagir avec le monde physique:
 - ... EN ENTRÉE: Collecter des informations du milieu physique, chimique, géographique, ...
 - ... EN SORTIE: Effectuer des actions sur ces milieux comme allumer un ampoule DEL, faire tourner un moteur
- Équipés pour cela de certaines caractéristiques des ordinateurs:
 - Unité de traitement
 - Mémoire morte, mémoire vive, mémoire flash, bus d'entrées et de sorties
- Pourvus de bornes auxquelles vous branchez des sondes ou circuits de détections
- Communications sans fil dans certains modèles



<https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino>



Introduction aux microcontrôleurs

- Les circuits électroniques sont sensibles :
 - À l'électricité statique
 - Touchez une pièce de métal avant de manipuler un circuit électronique
 - En cas de décharge, les composants électroniques, dont le microcontrôleur, peuvent être endommagés pouvant aller jusqu'à la destruction totale ou partielle du circuit
 - Aux surcharges électriques
 - Calculez la tolérance d'un circuit pour la bonne valeur des pièces à brancher ensemble
 - Dépasser les limites peut avoir les mêmes conséquences que l'électricité statique
 - Aux erreurs de manipulation ou de connexion
 - Maintenir votre table de travail propre et ordonnée
 - **BIEN DOCUMENTER** et dessiner le circuit selon vos calculs
- **SURTOUT**, assurez votre protection selon les directives

Introduction aux microcontrôleurs

- Dans ce cours, deux modèles seront utilisés :
 - Arduino pour les premiers montages qui seront reliés à un ordinateur par un câble USB
 - ESP32 pour les montages communicants sans fil et pouvant être relié à internet
- ESP32 peut servir à l'internet des objets connectés (IOT : Internet Of Things)

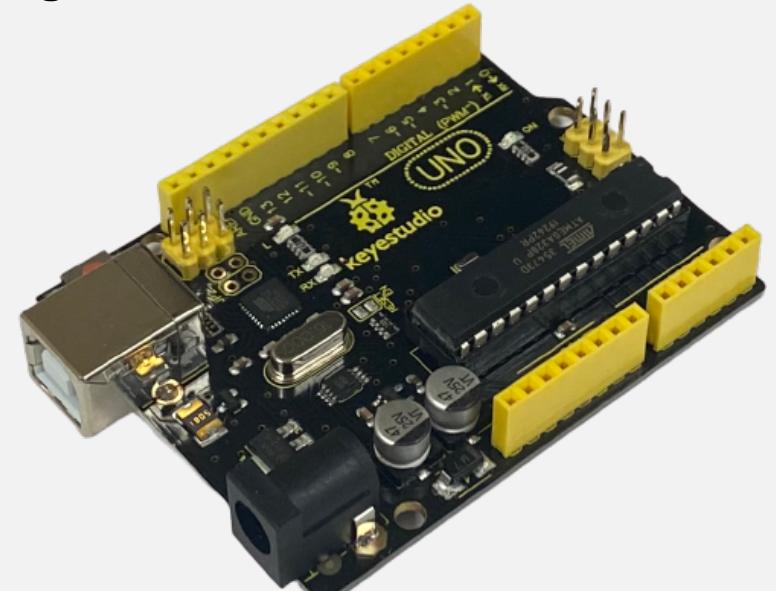
Votre kit Arduino

- Coffret contenant une variété de pièces et détecteurs permet d'expérimenter et assembler des prototypes :
 - Microcontrôleur Arduino UNO
 - Platine d'essai (« BreadBoard »)
 - Câble USB de raccordement à l'ordinateur
 - Afficheur LCD 16 X 2
 - Afficheurs 7-segments: unitaire et quadruple
 - Boutons poussoirs
 - Résistances électroniques de valeurs différentes
 - Nombreuses DELs de couleurs différentes
 - Fils de raccordements
 - Nombreux détecteurs: mouvement, sonore, pression



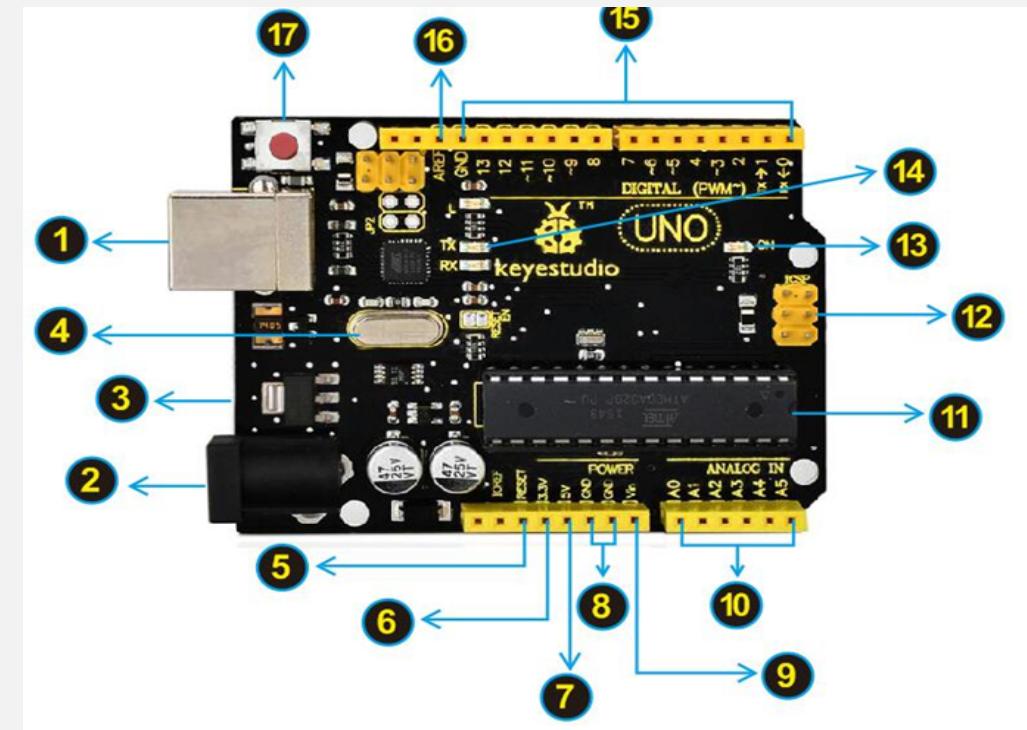
Le microcontrôleur Arduino

- Pourquoi Arduino ?
 - Très populaire, à prix abordable, facile à configurer et à programmer
 - Alimentée par une faible tension (5 V / USB, 7-12 V)
 - Programmée avec le langage C populaire
- Ce qu'il contient ?
 - Microprocesseur ATmega328P
 - Mémoire vive : 2 Ko (SRAM)
 - Mémoire flash : 32 Ko (votre programme)
 - EEPROM : 1 Ko (persistance de données)
 - Bornes d'entrée et sortie pour un branchement commode et réutilisable
 - Permet donc le prototypage de projets pour objets connectés



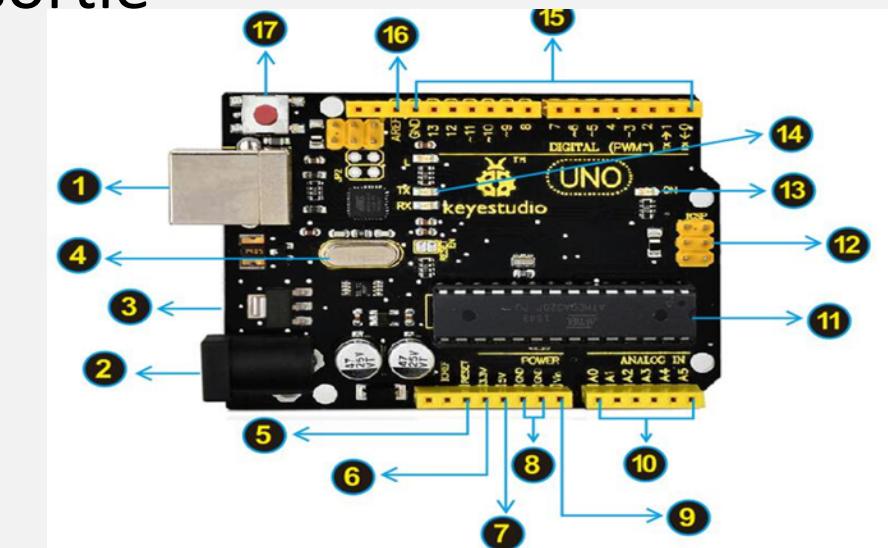
Architecture d'Arduino UNO (1/3)

- 1 : port USB: alimentation ou téléchargement du code binaire
- 2, 9 : alimentation externe
 - 2: connecteur 9V
 - 9: branchement à fabriquer!
- 3 : Régulateur de tension
- 4 : cristal à 16 MHz
- 5 et 17: « reset »
- 6,7 : bornes d'alimentation externe
 - 6 délivre 3,3 V
 - 7 délivre 5 V
- 10 : bornes d'entrées analogiques
- Référence : https://wiki.keyestudio.com/Getting_Started_with_Arduino



Architecture d'Arduino UNO (2/3)

- 11 : Atmega 328P (« MCU »)
- 13 : Indicateur de l'alimentation
- 14 : 2 DELs montrant l'activité du port série (Tx et Rx)
- 15 : Bornes programmables en entrée OU en sortie
- 8 : Masse (« ground »)
- 16 : Analog Reference (AREF)



- Référence : https://wiki.keyestudio.com/Getting_Started_with_Arduino

Architecture Arduino (3/3)

- Caractéristiques physiques de l'Arduino UNO

Catégorie	Valeur
Microcontrôleur	ATmega 328
Fréquence d'horloge	16 MHz
Ports numériques entrées ou sorties	14 dont 6 commutables en PWM
Ports-analogiques	6 entrées analogiques
Mémoires	32 Ko Flash, 2Ko SRAM, 1Ko EEPROM
Chargeur d'amorçage	0,5 Ko (en mémoire Flash)
Interface pour PC	USB type A-B
Dimensions	6,86 cm X 5,3 cm

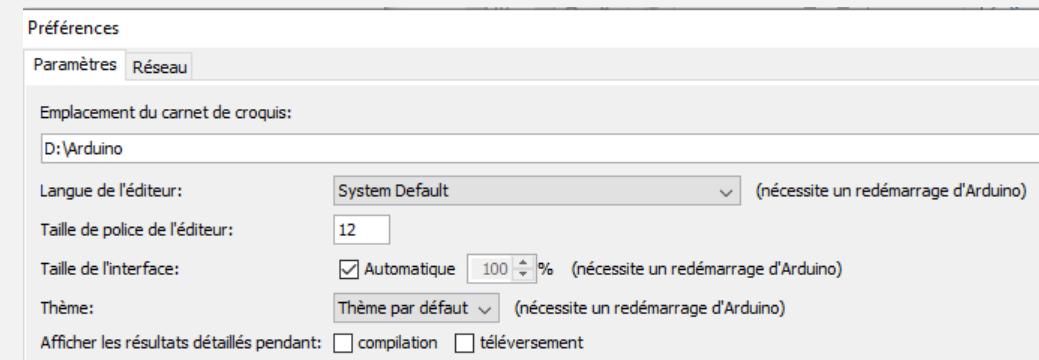
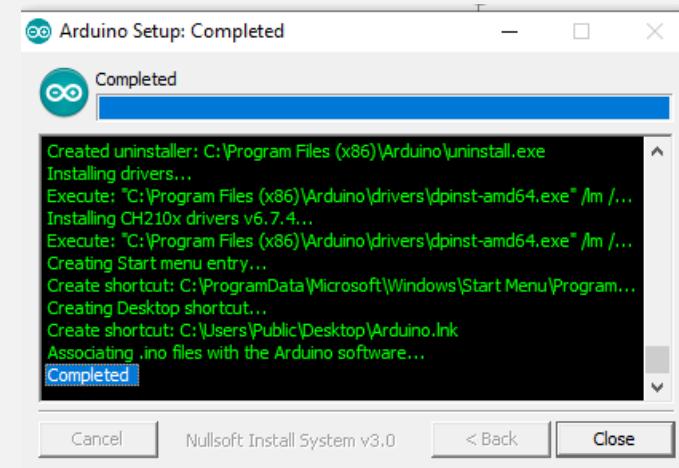
Catégorie	Valeur
Tension de service	5 V
Alimentation extérieure (recommandée)	7-12 V
Alimentation extérieure (limites)	6- 20 V
Tolérance maxi. par broche d'E/S (en c.c.)	20 mA; limite totale de la carte 200 mA

Organisation d'un atelier

- Procédures pour faciliter le travail et éviter les accidents
 - Dérouler le tapis de protection sur la table
 - Au début d'un exercice, étalez uniquement les pièces nécessaires : éviter les erreurs et faciliter votre dépannage
 - Évitez de placer les outils métalliques à proximité de la surface de travail
 - L'usage de la loupe permet d'identifier les couleurs des résistances et les codes des pièces
 - L'usage de pinces, de la gomme à papeterie ou du ruban permet de maintenir le matériel en place

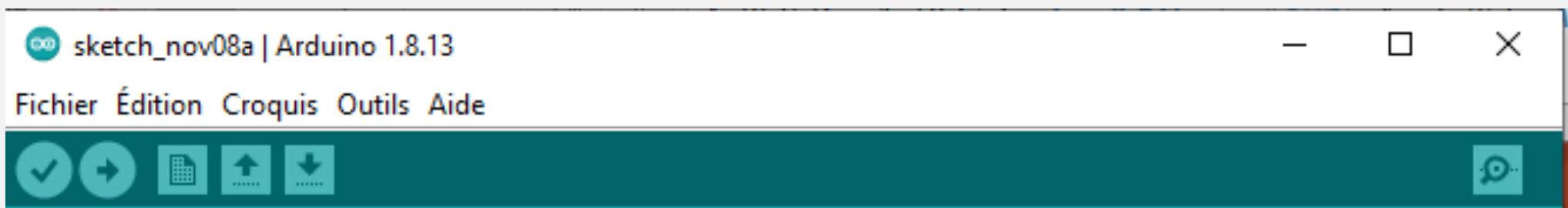
Arduino IDE

- Environnement de développement intégré (IDE)
- Téléchargement à partir du site web Arduino.cc
- Installation sur portable ou support externe (USB, SSD)
- Paramètres disponibles:
 - répertoire servant de dépôt pour vos programme
 - langue de l'interface
 - Éditeur externe
 - ...



Arduino IDE

- Principaux menus de fonctions:
 - Configurer le pilote et le port USB des cartes utilisées
 - Gérer les programmes
 - Charger, éditer, Sauver,
 - Compiler, téléverser le code binaire dans le microcontrôleur
 - ...



Structure d'un « sketch »

- Un programme Arduino se nomme un sketch;
- Le langage C++ est choisi par défaut
- Deux fonctions principales OBLIGATOIRES
 - **setup()** { instructions }
 - Exécutée une seule fois au démarrage de l'application
 - **Loop()** { instructions }
 - Les instructions sont exécutées en séquence et répétées en boucle,
 - interrompue par une rupture du courant OU
 - par l'appui du bouton « Reset »



The screenshot shows the Arduino IDE interface with the title bar "sketch_nov06a | Arduino 1.8.13". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar has icons for file operations like Open, Save, and Print. The code editor window displays the following code:

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
```

The status bar at the bottom right indicates "Arduino Uno sur COM3".

Structure d'un « sketch »

- Extension standard: **.ino**
- Sauvegardez vos sketch dans Git
- Fonction déclarée avec (*arguments*) ;
- instructions comprises entre { }
 - Instructions séparées par point-virgule ;
- 2 format de commentaires
 - Long : `/* Commentaire */`
 - Court : `// Commentaire sur une ligne`

```
sketch_Blink
1 /* plusieurs lignes de commentaires
2  * documente le sketch|
3 *
4 */
5 void setup() {
6     // put your setup code here, to run once:
7
8 }
9
10 void loop() {
11    // put your main code here, to run repeatedly:
12
13 }
```

Structure d'un « sketch »

- Déclarations de variables
 - type nom [= valeur]; // le point virgule est requis
 - Exemple int led = 13; // plusieurs types disponibles
- Appel de fonctions
 - nom([constante / variable [, constante / variable]])
 - Exemple pinMode(led, OUTPUT)
- Pour les reconnaître, ...
 - . . . l'IDE colore les noms valides
 - . . . les constantes « internes » inscrites en majuscules
- Documentation des fonctions Arduino : <https://www.arduino.cc/reference/en/>

```
digitalWrite (led, HIGH);  
delay(1000);
```