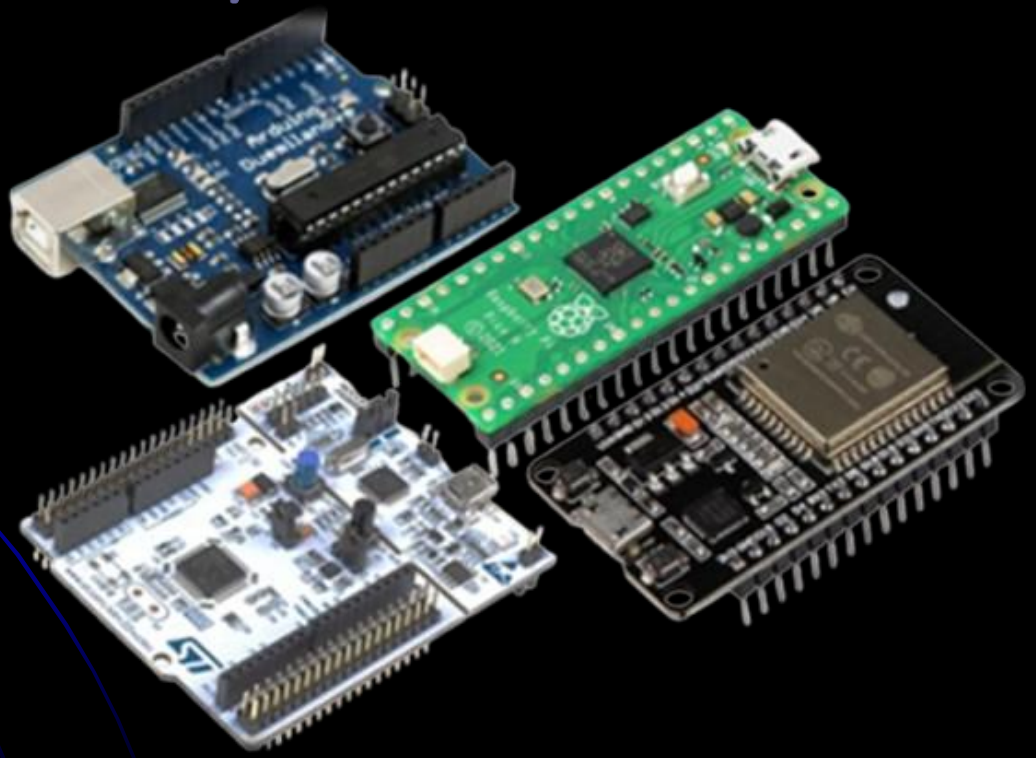
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE LA FORMATION ET DE L'ENSEIGNEMENT PROFESSIONNELS

IFEP SBA

MICROCONTROLEURS

وحدة التحكم الدقيق



Présenté par :

HALAILI Mohamed

MICROCONTRÔLEUR

Un **microcontrôleur** est un micro-ordinateur conçu pour effectuer des tâches spécifiques au sein d'un système embarqué. Il s'agit d'un composant monobloc, composé :

- Unité centrale de traitement (CPU),
- Mémoire vive (RAM),
- Mémoire morte (ROM),
- Ports d'E/S,
- Registres de fonctions spécifiques,
- Temporisateurs,
- Oscillateur à quartz,
- Convertisseur numérique-analogique (CNA)
- Convertisseur analogique-numérique (CAN),

Le tout dans une seule puce.

Les familles de microcontrôleurs

Les familles de microcontrôleurs regroupent des composants partageant une architecture similaire (8, 16 ou 32 bits) et des périphériques communs. Les plus répandues incluent ARM Cortex-M, PIC, AVR (utilisés dans Arduino), STM32, et ESP32, spécialisées pour les systèmes embarqués, l'IoT ou le contrôle simple, caractérisées par leur faible consommation.

Cartes de développements pour microcontrôleur

Une **carte de développement** pour microcontrôleur est une carte électronique conçue pour programmer, **tester** et **réaliser** des projets électroniques facilement autour d'un microcontrôleur.

Cartes de développements pour microcontrôleur

Une carte de développement contient généralement :

- Un microcontrôleur (ESP32, Arduino, STM32, PIC…)
- Un port USB pour la programmation
- Des broches GPIO (entrées/sorties)
- Une alimentation électrique
- Parfois elle contient :
  - Wi-Fi / Bluetooth
  - Convertisseur USB-Série
  - Boutons poussoirs
  - LEDs de test
  - Régulateur de tension

Cartes de développements pour microcontrôleur

Plusieurs cartes microcontrôleurs sont réputées pour leurs caractéristiques uniques. Parmi elles, les plus populaires sont:

Les cartes Arduino

Les cartes **Arduino** sont des cartes de développement électroniques basées sur des microcontrôleurs **Atmega**.

Les cartes Arduino

Elles sont conçues pour apprendre facilement :

- la programmation
- l'électronique
- les systèmes embarqués
- la robotique
- l'automatisation

Les cartes Arduino

Elles sont conçues pour apprendre facilement :

- la programmation
- l'électronique
- les systèmes embarqués
- la robotique
- l'automatisation

Ces cartes de développement prennent en charge le langage de programmation C/C++ et MicroPython sur certaine cartes

Les cartes Raspberry

Les cartes **Raspberry** sont des cartes électroniques programmables ,elles utilisent surtout des microprocesseurs ARM , elles sont concus pour :

- l'informatique embarquée ,
- l'apprentissage de la programmation
- l'IoT
- la robotique
- l'automatisation
- les projets industriels

Les cartes Raspberry

La plupart des Raspberry sont de petits ordinateurs complets. Elles peuvent être programmés avec :

- Python
- C/C++
- MicroPython
- Scratch
- JavaScript

Les cartes ESP32

Les cartes **ESP32** sont des cartes de développement basées sur le microcontrôleur ESP32

Elles sont très utilisées dans :

- ✓ l'IoT (Internet des objets)
- ✓ les systèmes embarqués
- ✓ la domotique
- ✓ l'électronique industrielle
- ✓ les projets connectés Wi-Fi

Elles sont populaires parce qu'elles intègrent directement :

- ✓ le Wi-Fi
- ✓ le Bluetooth
- ✓ un microcontrôleur puissant

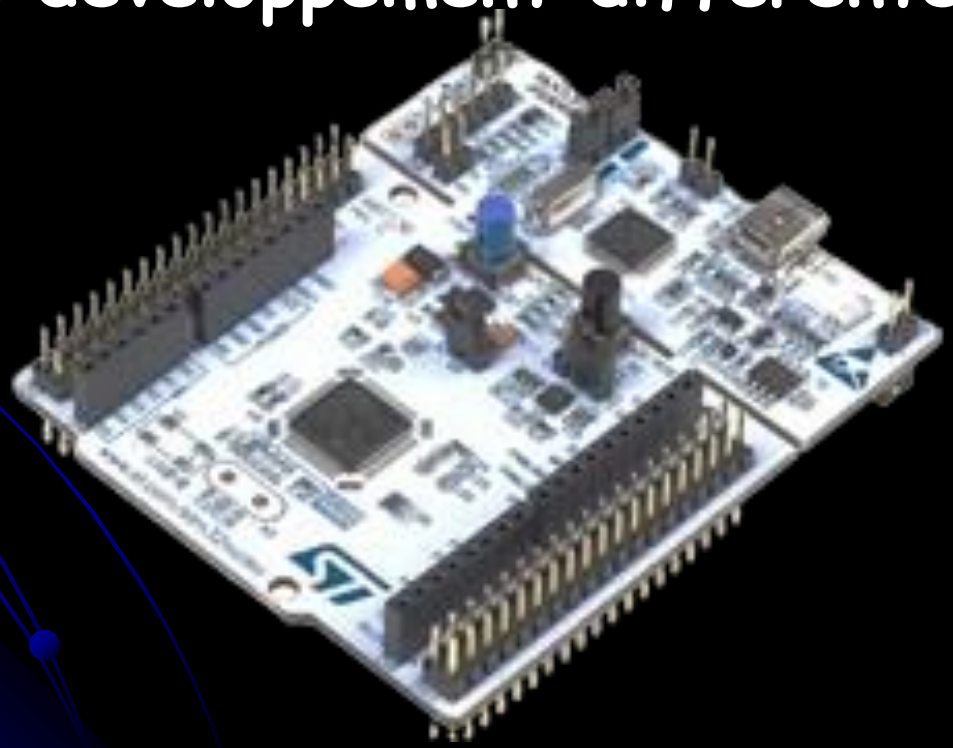
Les cartes ESP32

Ces cartes de développement prennent en charge le langage de programmation C/C++ et Python (MicroPython)

Les cartes STM32 Nucleo

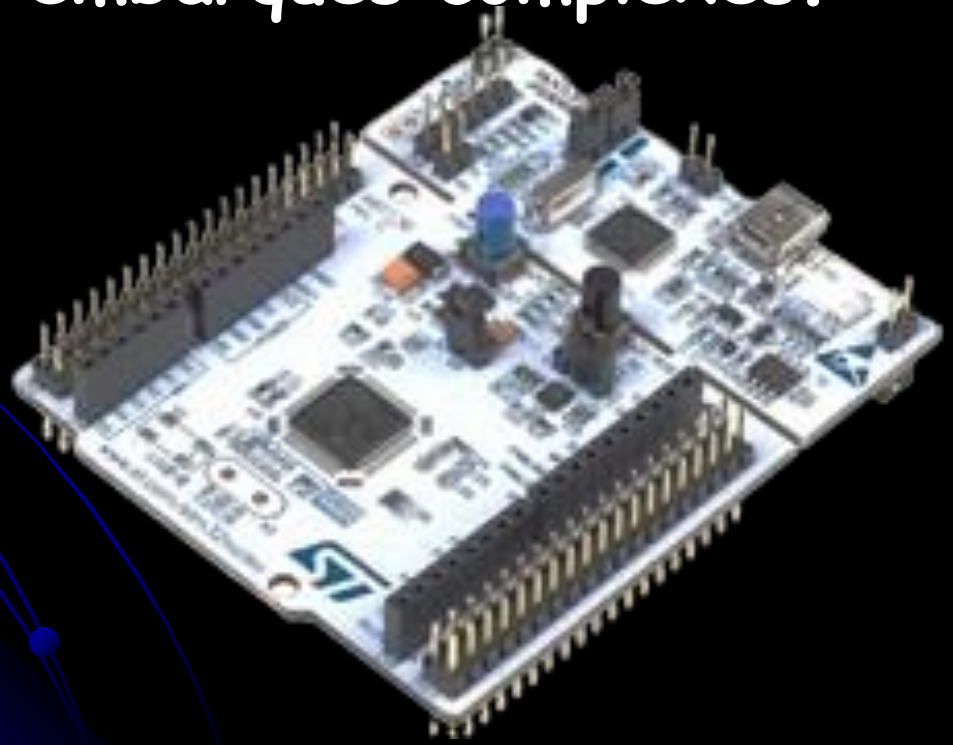
Les cartes **STM32** existent en très grand nombre.

La famille **STM32** contient aujourd'hui plus de **20 grandes séries** de microcontrôleurs et des centaines de cartes de développement différentes.



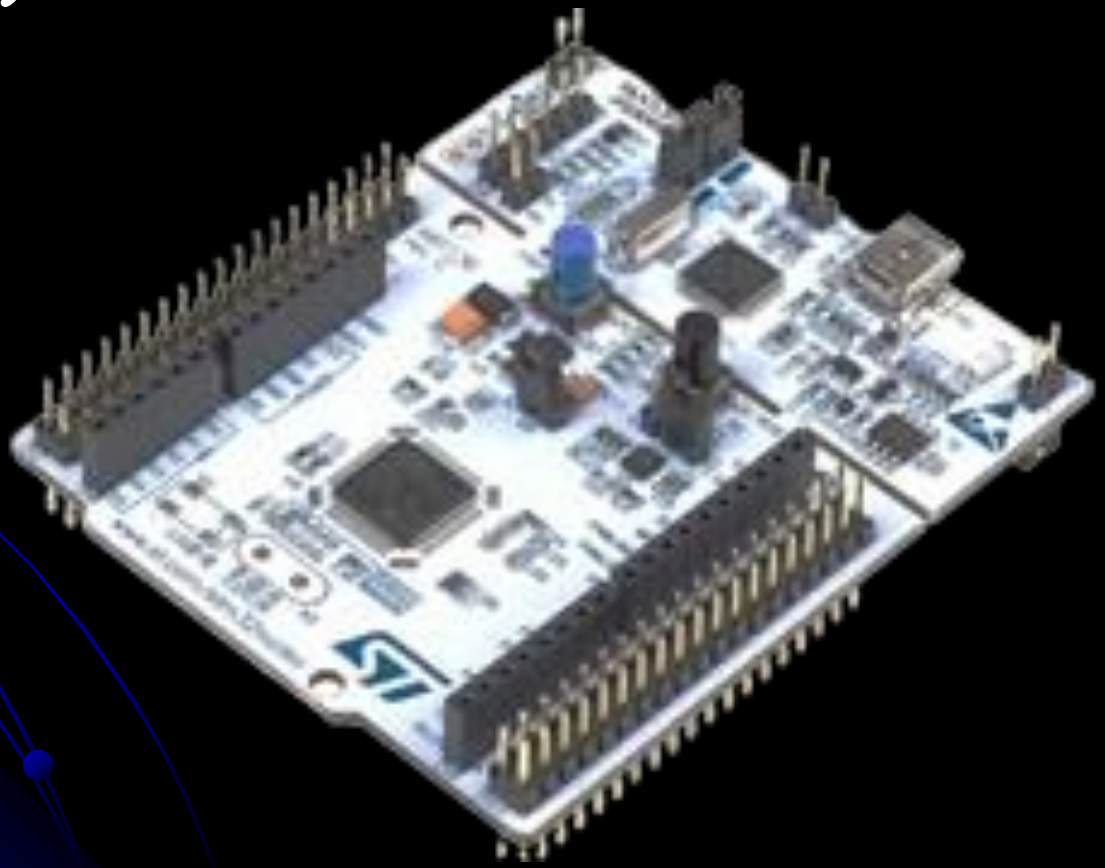
Les cartes STM32 Nucleo

Les cartes de développement **STM32** sont utilisées pour le développement de **projets électroniques haut de gamme**, tels que l'automatisation industrielle et les systèmes embarqués complexes.



Les cartes STM32 Nucleo

Ces cartes de développement prennent en charge le langage de programmation C/C++. et Python (MicroPython)



INTRODUCTION

Arduino est un projet créé par une équipe de développeurs. C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes.

INTRODUCTION

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, nous allons programmer des systèmes électroniques.

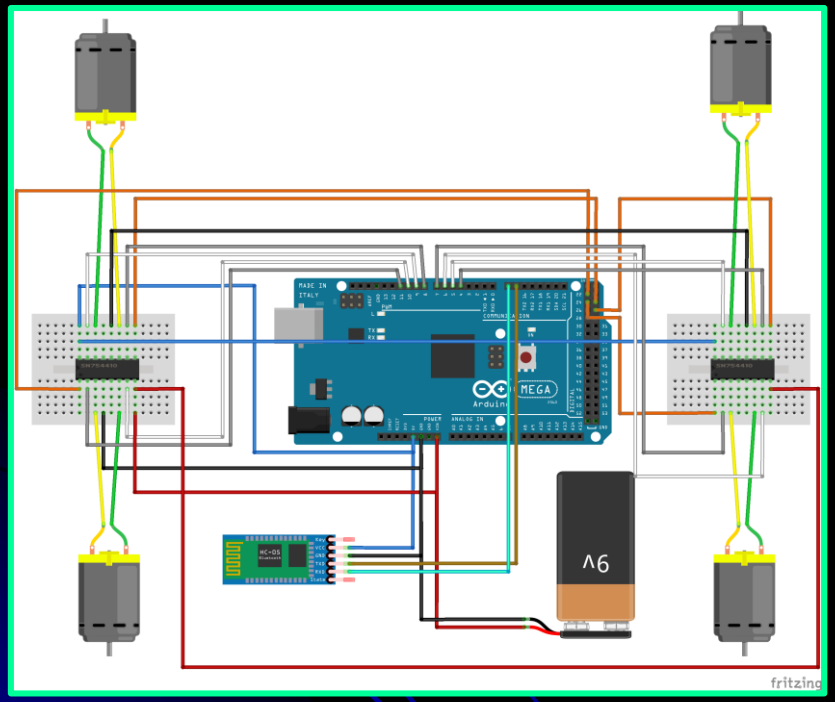
INTRODUCTION

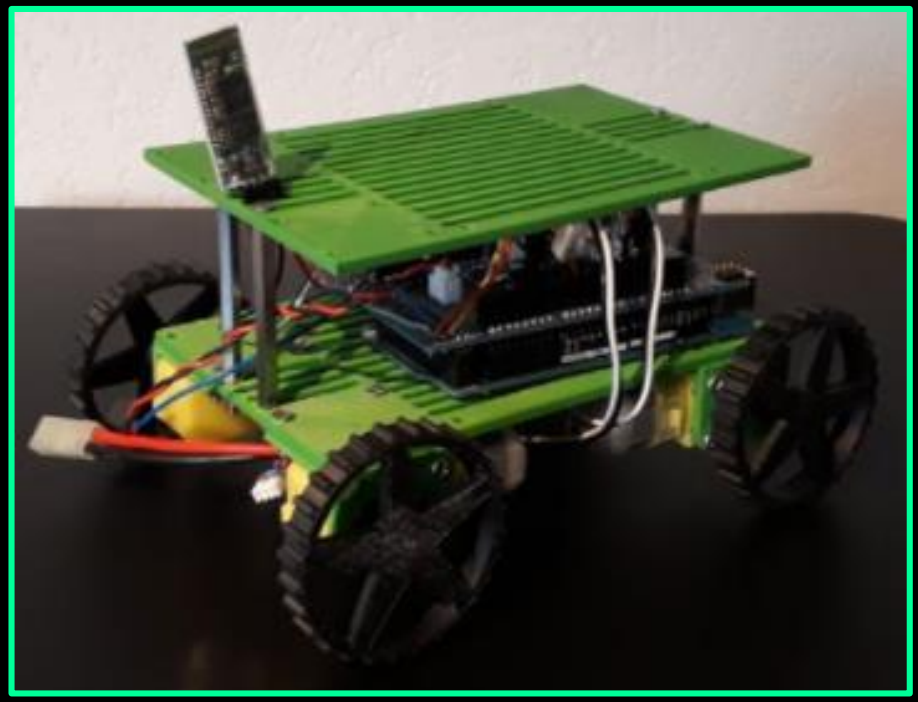
Le gros avantage de l'électronique programmée c'est qu'elle **simplifie** grandement les schémas électroniques et par conséquent, le **coût** de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

Le **système Arduino** nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines , l'étendue de l'utilisation de l'Arduino est gigantesque. Voici quelques **exemples** de ce que vous pouvez réaliser avec une telle carte :

Exemples de réalisation Ardouino

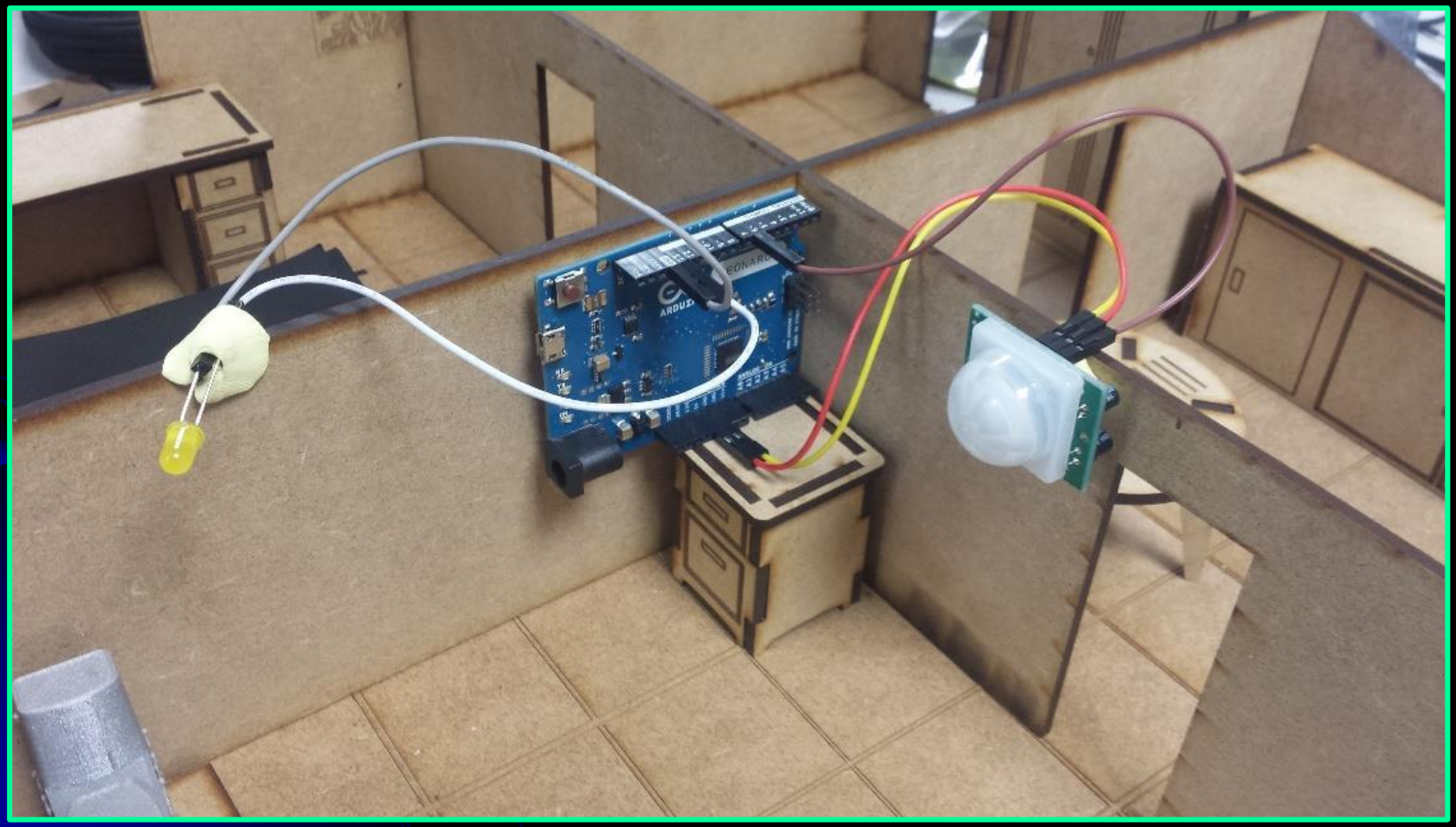
Un robot mobile capable d'éviter les obstacles ou de suivre une ligne au sol





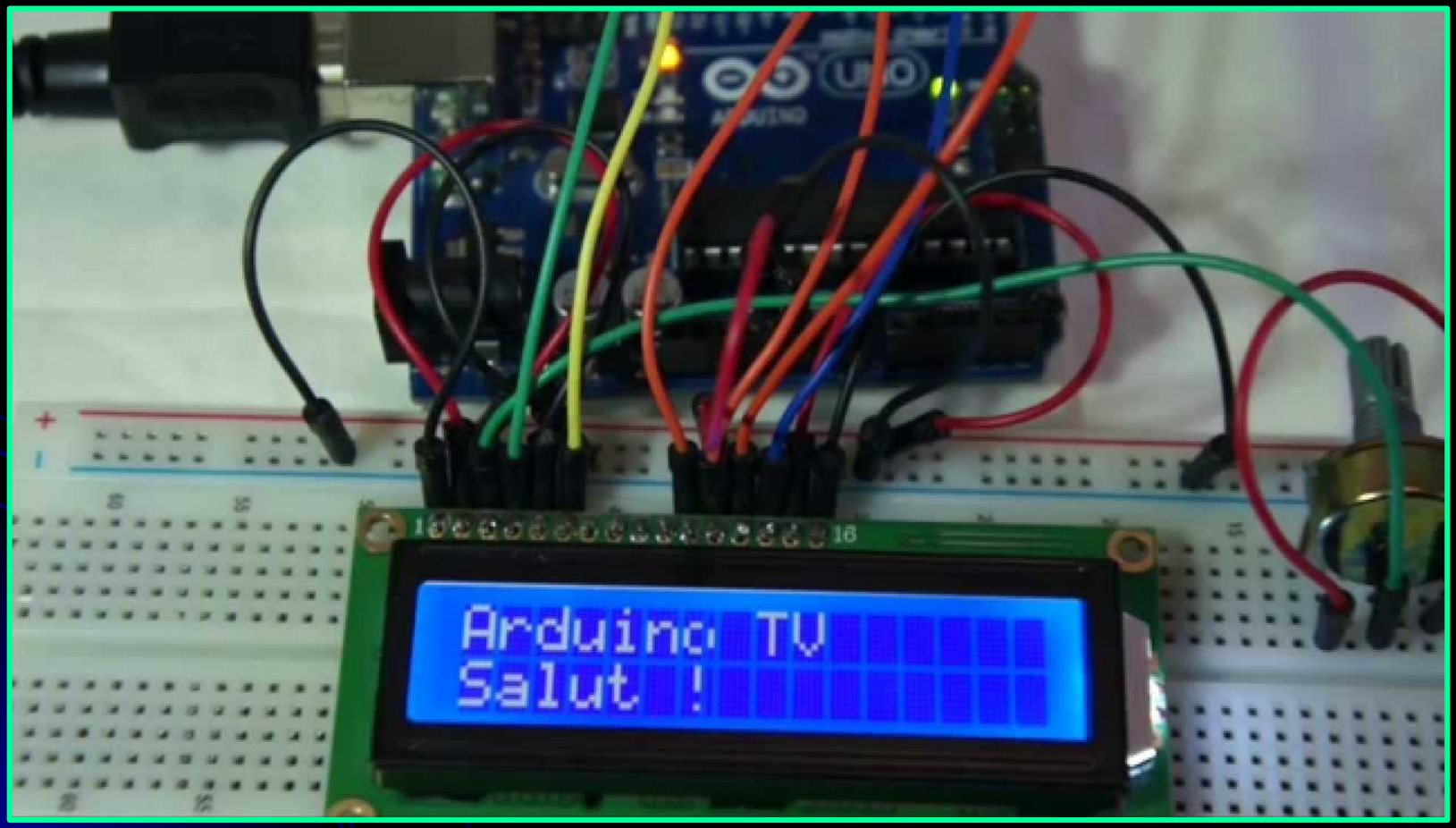
Exemples de réalisation Ardouino

Une interface entre votre téléphone mobile et les éclairages de votre maison



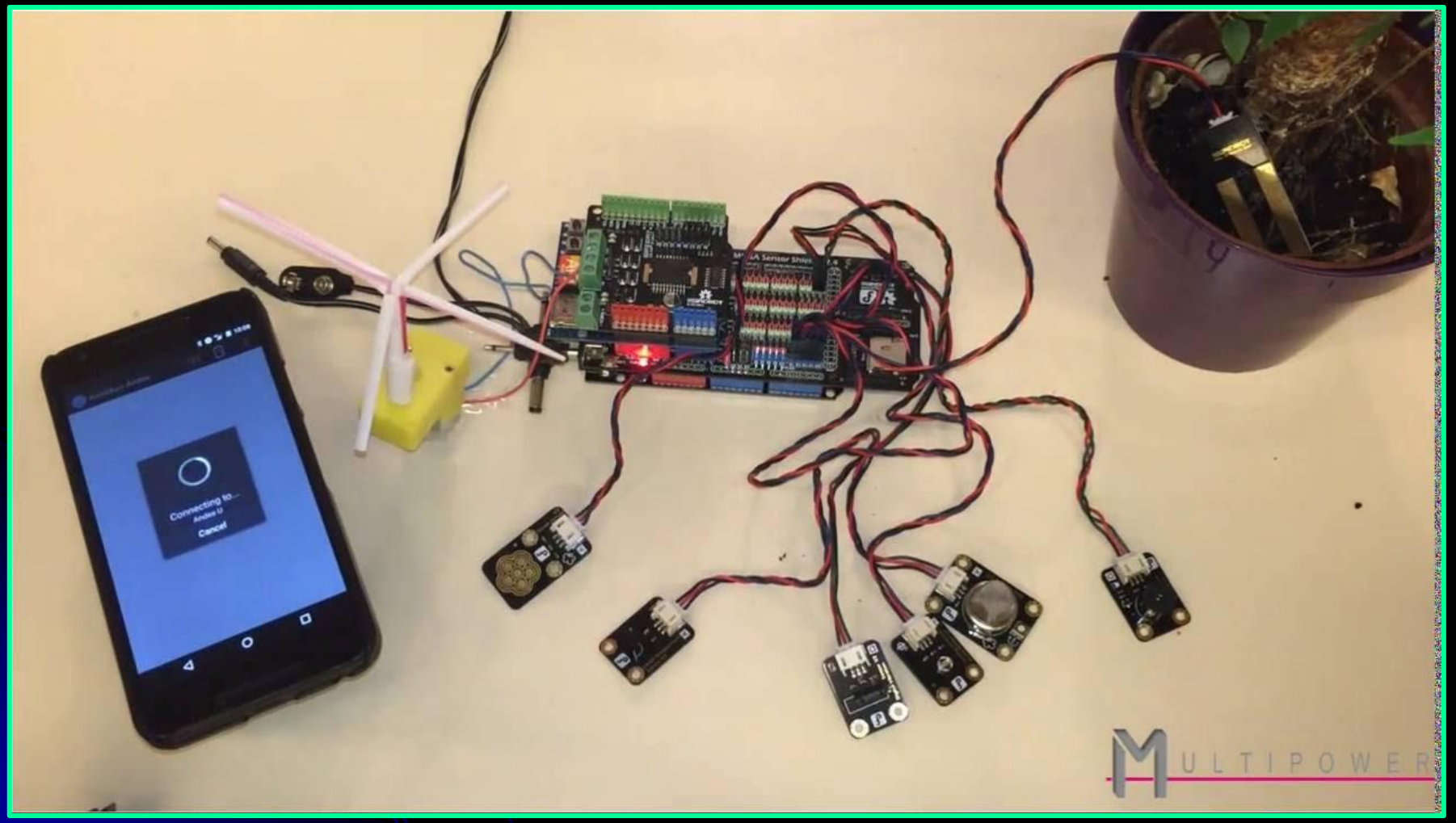
Exemples de réalisation Ardouino

Des afficheurs d'informations à base de textes défilants sur des panneaux à LEDs



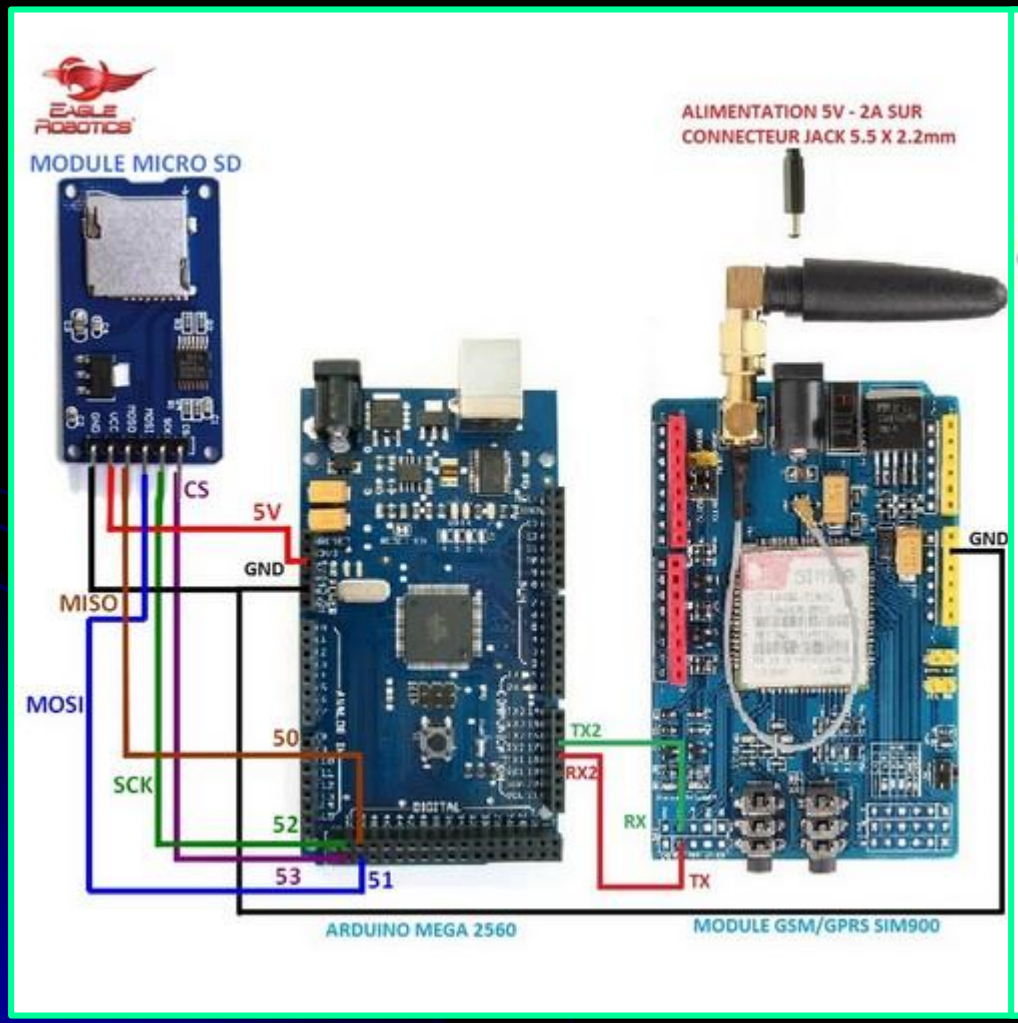
Exemples de réalisation Ardouino

Une station météorologique consultable sur le Web



Exemples de réalisation Ardouino

Un pilote de caméra de surveillance par smartphone



Exemples de réalisation Ardouino

Drone de surveillance

Présentation de l'Arduino

L'**Arduino** est une plateforme open source d'électronique programmée qui est basée sur une carte à **microcontrôleur** et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable

Présentation de l'Arduino

La programmation se fait à l'aide d'un langage proche du C/C++, dont les bases sont faciles d'accès. Le logiciel nécessaire fonctionne à la fois sur Mac OSX, Windows et GNU/Linux et demande très peu de ressources.

Principe de fonctionnement de l'Arduino

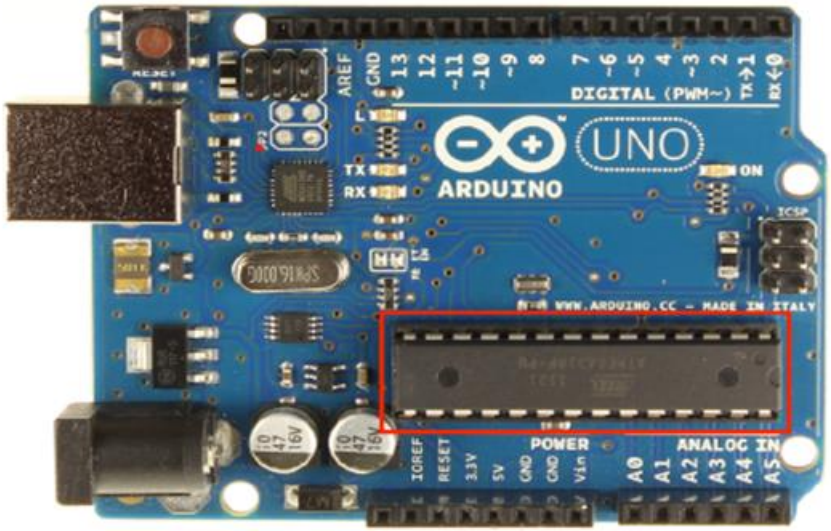
1. On réalise le programme sur un ordinateur.
2. On connecte l'ordinateur à l'Arduino via une prise USB.
3. On envoie le programme sur l'Arduino.
4. L'Arduino exécute enfin le programme de manière autonome.

Présentation de l'Arduino

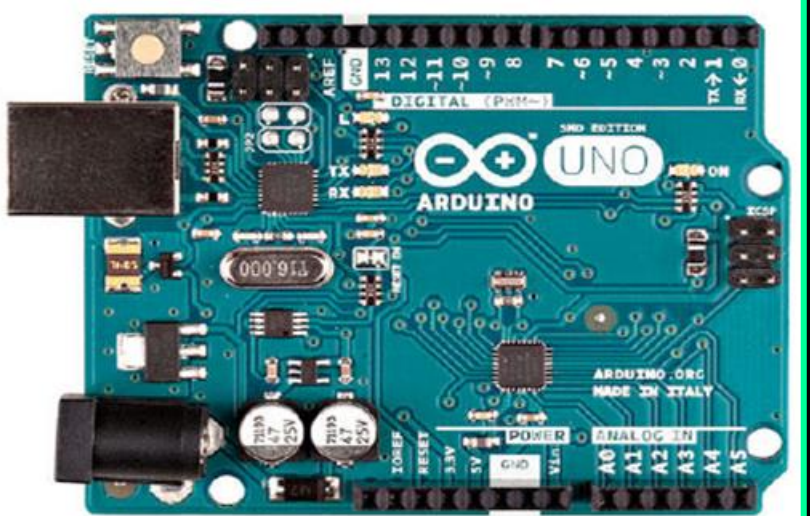
Il existe deux modèles d'Arduino Uno: l'un avec un microcontrôleur de grande taille

, et un autre avec un microcontrôleur dit **SMD** (SMD: Surface Mounted Device, soit composants montés en surface, il n'y a pas de différence entre les deux types de microcontrôleurs.

Arduino Uno



Arduino Uno SMD



Quelques cartes Arduino

Arduino Uno

Arduino Leonardo

Arduino Due

Arduino Yún

Arduino Tre

Arduino Micro

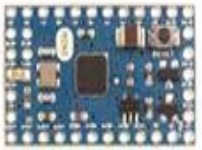
Arduino Robot

Arduino Esplora

Arduino Mega ADK

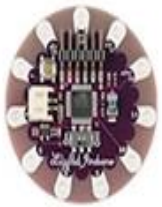
Arduino Ethernet

Arduino Mega 2560



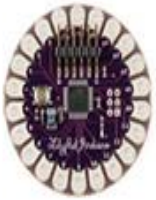
Arduino Mini

LilyPad Arduino USB



LilyPad Arduino Simple

LilyPad Arduino SimpleSnap



LilyPad Arduino

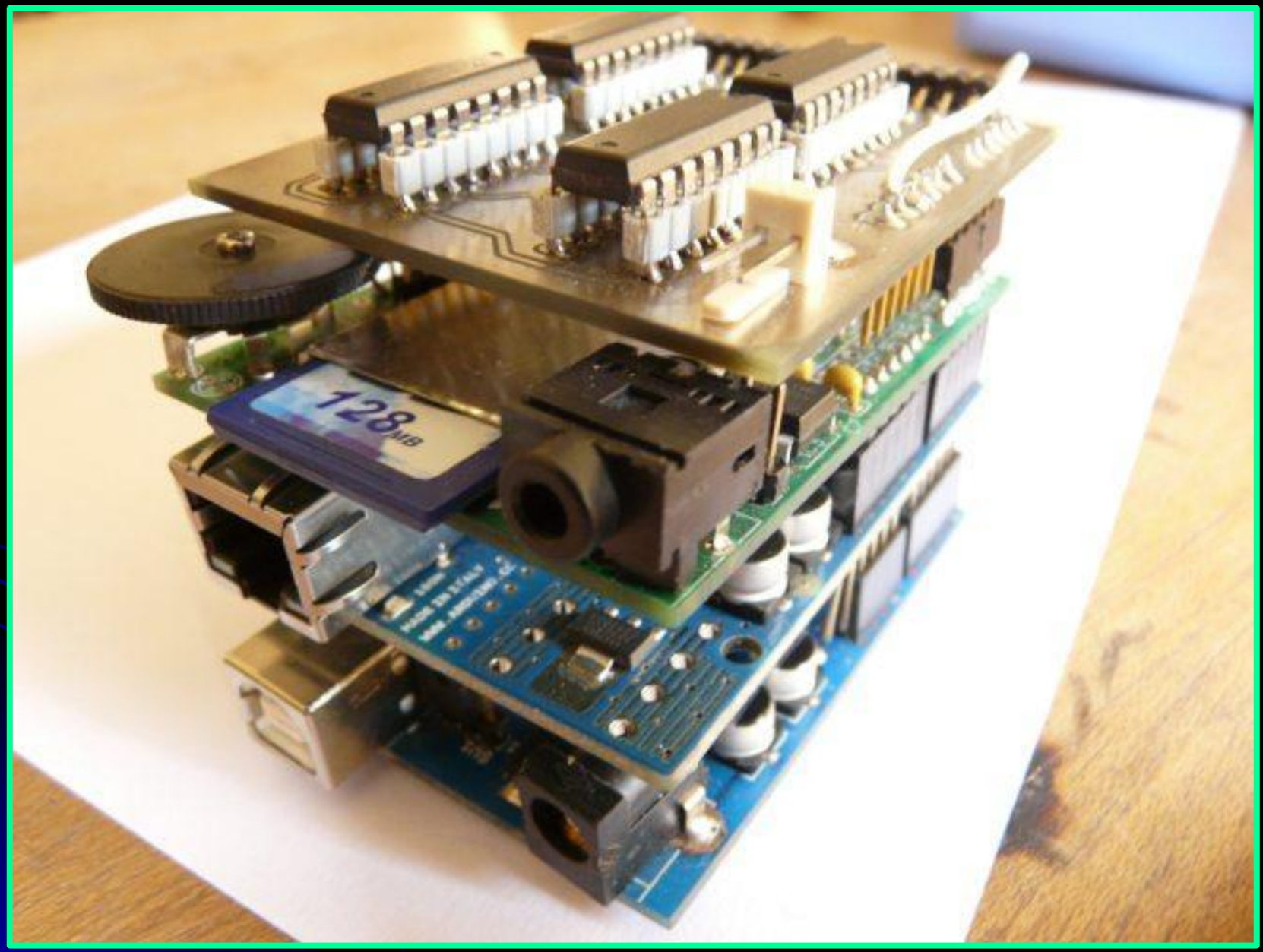
Arduino Nano

Arduino Pro Mini

Circuits additionnels à Arduino

Il est possible de spécialiser la carte Arduino en l'associant avec des **circuits additionnels** que l'on peut fabriquer soi-même ou acheter déjà montés.
Lorsqu'ils se branchent directement sur la carte, ces circuits s'appellent des « **shields** » ou cartes d'extension. Ces circuits spécialisés apportent au système des fonctionnalités diverses et étendues dont voici quelques exemples :

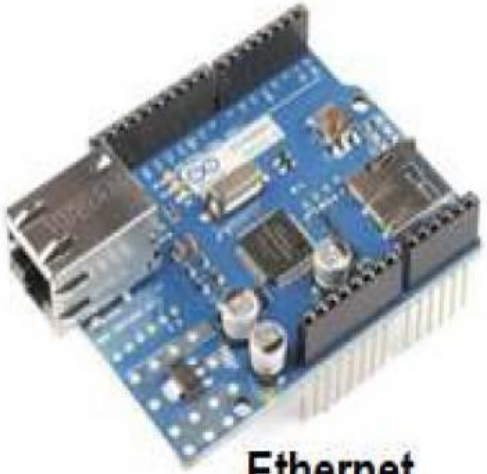
Circuits additionnels à Arduino



Quelques cartes Arduino

- Ethernet : communication réseau ;
- Bluetooth ou zigbee : communication sans fil ;
- Pilotage de moteurs (pas à pas ou à courant continu) ;
- Pilotage de matrices de LED : pour piloter de nombreuses LED avec peu de sorties ;
- Ecran LCD : pour afficher des informations ;
- Lecteur de carte mémoire : lire ou stocker des données ;
- Lecteur de MP3 ;
- GPS : pour avoir une information de position géographique ;
- joystick ;
- etc.

Quelques cartes Arduino



Ethernet

GSM

WIFI

Relais

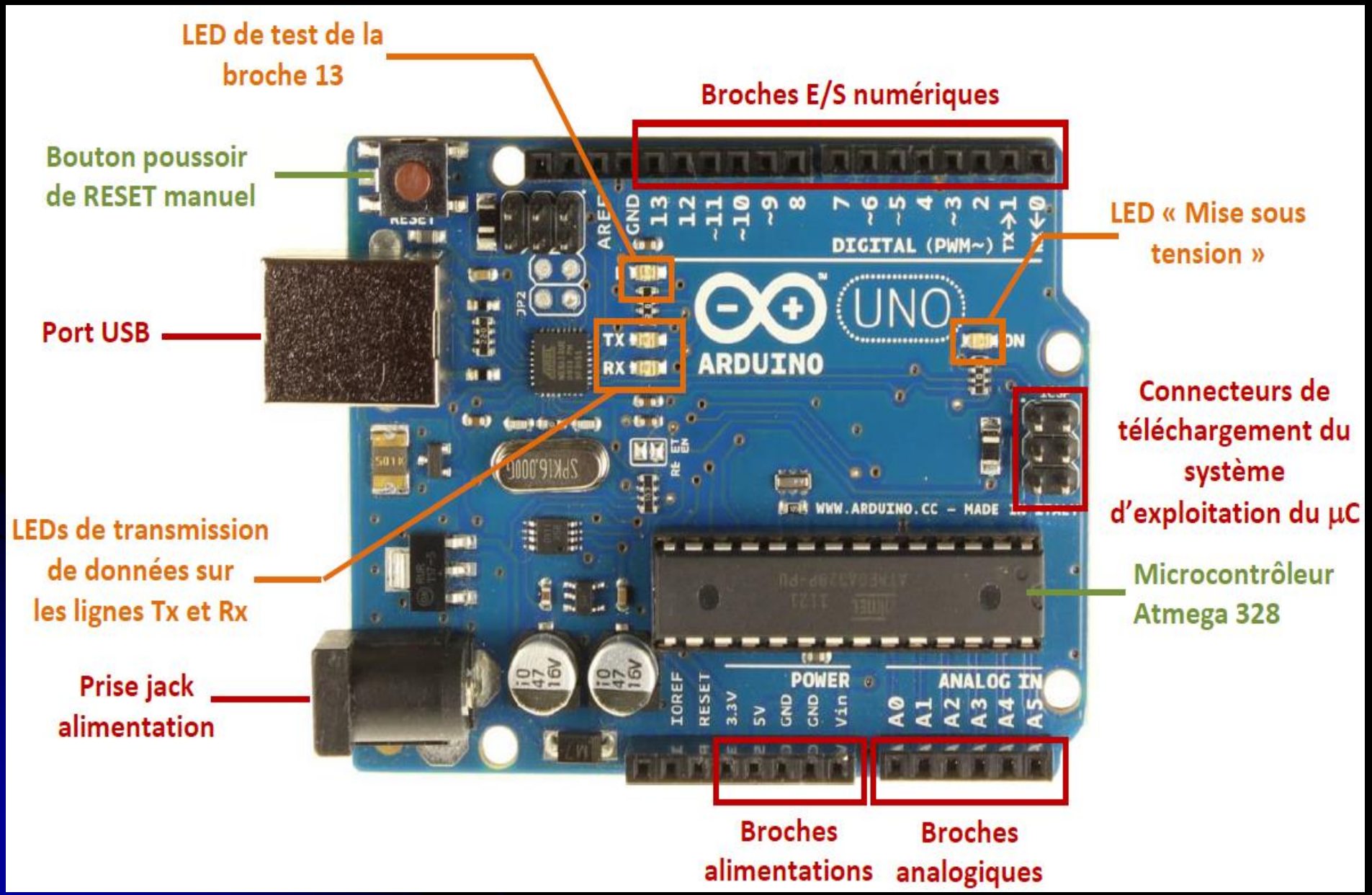
LCD

Commande moteurs

Arduino Uno

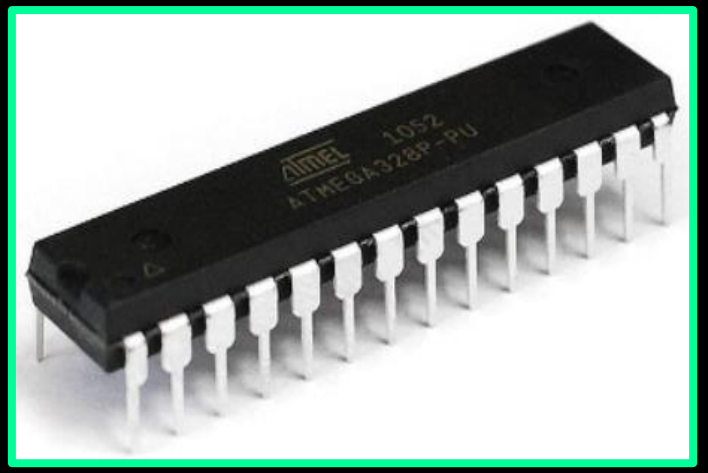
Il existe plusieurs types de cartes Arduino. La carte **Arduino Uno** est une des versions majeures des cartes Arduino. Il s'agit d'une carte au format « standard » Arduino c'est-à-dire environ 52 mm sur 65 mm. .

Schéma d'une platine Arduino Uno



Le microcontrôleur ATmega328

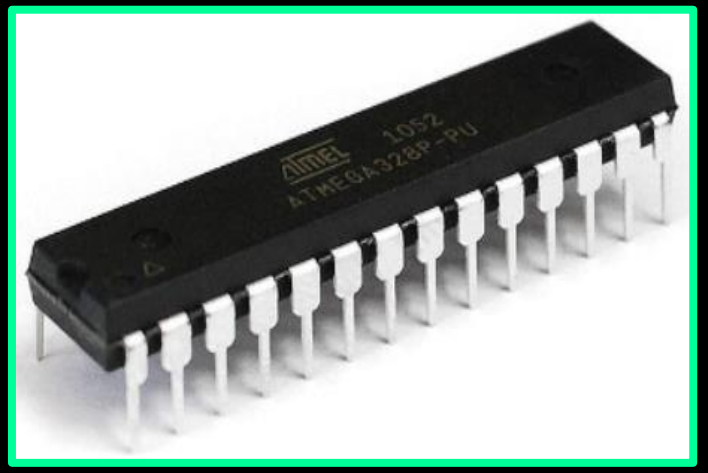
C'est le cerveau de notre carte. c'est un ATmega328, fabriqué par Atmel. Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter.



Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur, mettre en route ou arrêter un moteur…

Le microcontrôleur ATmega328

Un microcontrôleur est constitue par un ensemble d'eléments qui ont chacun une fonction bien déterminée. Il est en fait constitue des mêmes éléments que sur la carte mère d'un ordinateur :



Le microcontrôleur ATmega328

I. La mémoire

Il en possede 5 types :

- **La mémoire Flash** : C'est celle qui contiendra le programme a executer.Cette mémoire est effacable et re-inscriptible.

- **RAM** : c'est la memoire dite "vive", elle va contenir les variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du micro-controleur.

Le microcontrôleur ATmega328

- **EEPROM** : C'est le disque dur du microcontroleur. Vous pourrez y enregistrer des infos qui ont besoin de survivre dans le temps, meme si la carte doit etre arretee. Cette memoire ne s'efface pas lorsque l'on eteint le microcontroleur ou lorsqu'on le reprogramme.
- **Les registres** : c'est un type de memoire utilise par le processeur.
- **La memoire cache** : c'est une memoire qui fait la liaison entre les mémoires RAM et le microprocesseur,

Le microcontrôleur ATmega328

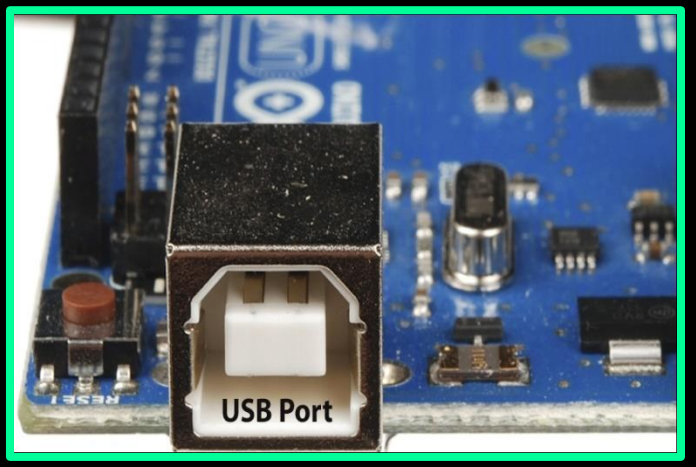
Le microprocesseur

C'est le composant principal du micro-controleur. C'est lui qui va exécuter le programme qu'on lui donnerons a traiter. On le nomme souvent le CPU. I la plusieur caractéristique les principaux sont :

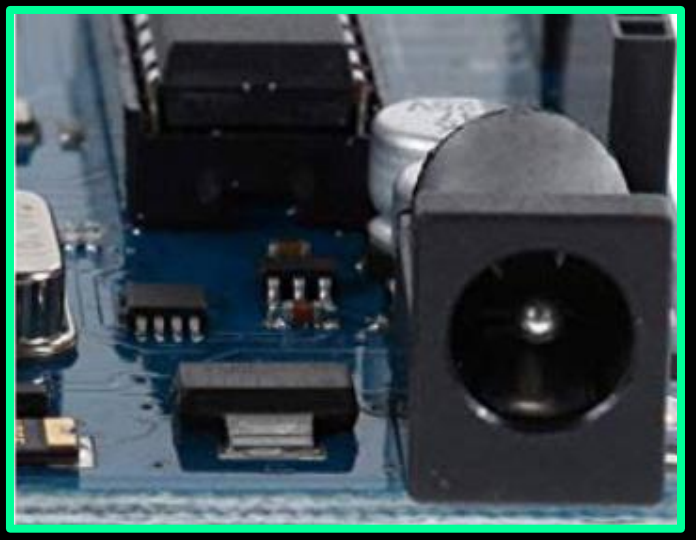
1. La fréquence de l'horloge qui définit la vitesse d'exécution des programme, elle est exprimer en Hertz (Hz)
2. Le bus de donnée (4bits, 8bits , 32bits ou 64bits)

Alimentation

Pour fonctionner, la carte a besoin d'une **alimentation**. Le microcontrôleur fonctionnant sous **5V**, la carte peut être alimentée en **5V** par **le port USB**



Elle peut aussi etre alimenter par une alimentation externe (Prise jack) qui est comprise entre 7V et 12V.



Alimentation

Cette tension d'alimentation doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte

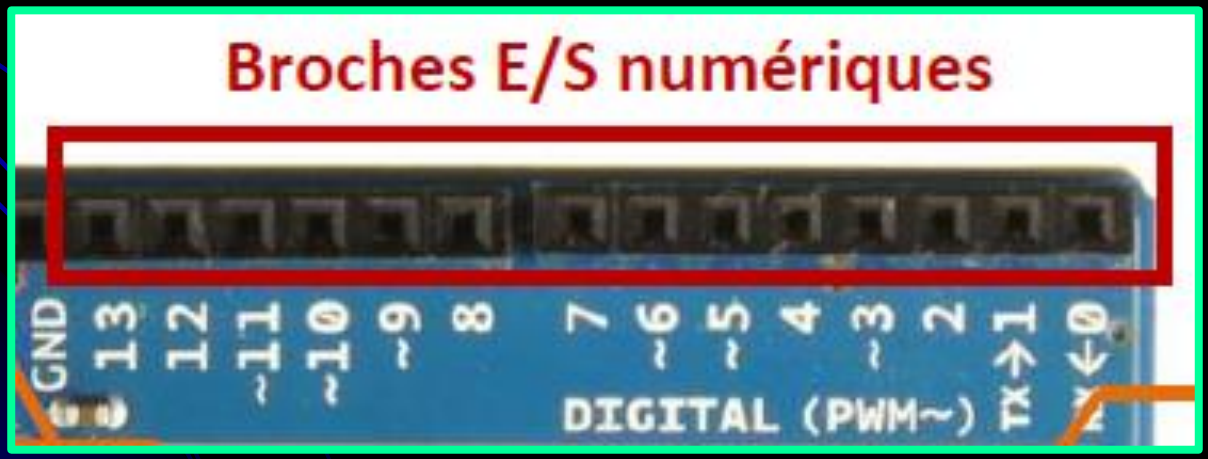
Les entrées/sorties

- La carte « Arduino Uno » dispose de:
- 14 E/S numériques
- 6 entrées analogiques

Les entrées/sorties

Entrées/sorties numériques

Chacune des 14 broches numériques (repérées 0 à 13) peut être utilisée en entrée (input) ou en sortie (output) sous le contrôle du programme.



Les entrées/sorties

Entrées/sorties numériques

Le sens de fonctionnement pouvant même changer de manière dynamique pendant son exécution.
Elles fonctionnent en logique TTL (0V-5V) ; chacune pouvant fournir (source) ou recevoir un courant maximal de 40 mA et dispose si besoin est d'une résistance interne de 'pull-up'

Les entrées/sorties

Entrées analogiques

Les 6 entrées analogiques, repérées A0 à A5 (PC0 à PC5), peuvent admettre toute tension analogique comprise entre 0 et 5 V (par défaut mais cela peut être modifié).

Visualisation

LED de test de la broche 13

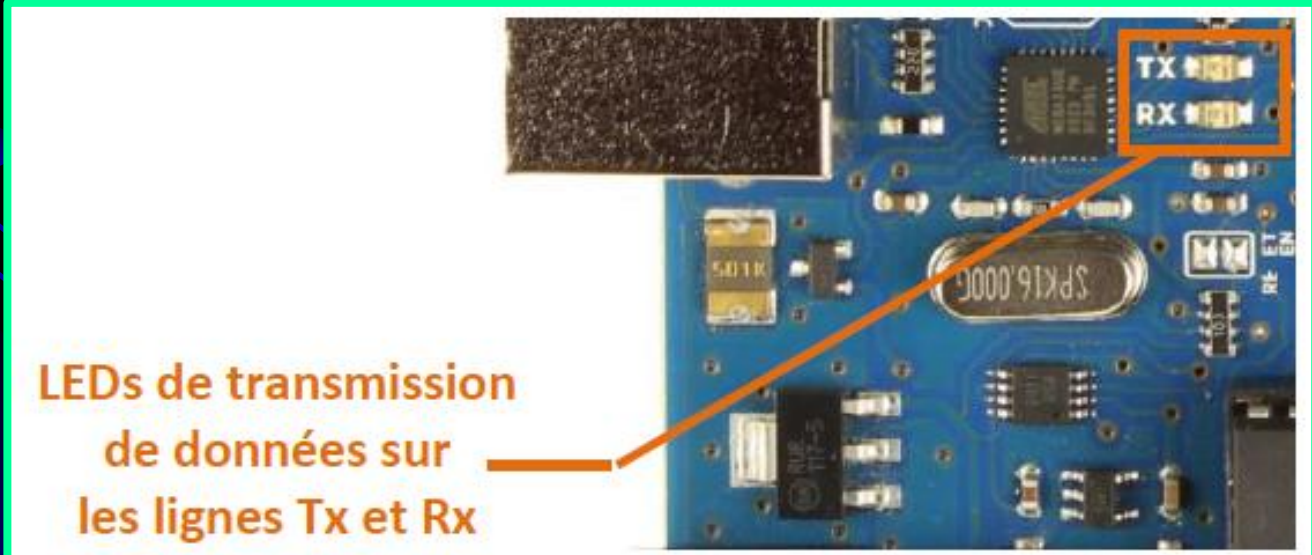
Celle tout en haut du cadre : elle est connectée a une broche du microcontrôleur et va servir pour tester le matériel.

Nota : Quand on branche la carte au PC, elle clignote quelques secondes

Visualisation

LEDs de transmission de données

Les deux LED du bas du cadre : servent a visualiser l'activite sur la voie serie (une pour l'emission et l'autre pour la reception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement



Reset

Bouton poussoir
de RESET manuel

A la mise sous tension un **reset** automatique permet au programme contenu en mémoire du microcontrôleur de démarrer automatiquement dès que la carte Arduino est alimentée.
La carte« Arduino Uno » est également équipée d'un bouton poussoir de **reset** manuel. Un appui sur celui-ci permet de relancer l'exécution d'un programme si nécessaire, soit parce qu'il s'est « planté » soit tout simplement parce que l'on souhaite le faire repartir de son début

Caractéristiques principales de l'Ardouino Uno

Microcontrôleur (MCU)	ATmega328
Tension d'utilisation	**5V**
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (min, max)	6-20V
E/S digitales	**14 (dont 6 pour les sorties PWM)**
Entrées analogiques	**6**
Courant continu par pin (E/S)	**40 mA**
Courant total sur les 14 pins	**200 mA**
Courant DC de la pin 3.3 volts	50 mA
Mémoire Flash	32 KB (ATmega328) dont 0.5 KB utilisée par le bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Fréquence d'horloge	**16 MHz**
Longueur	68.6 mm
Largeur	53.4 mm
Poids	25 g

PWM : Pulse Width Modulation que l'on traduit en français par MLI (Modulation de Largeur d'Impulsion).

Le logiciel Arduino IDE

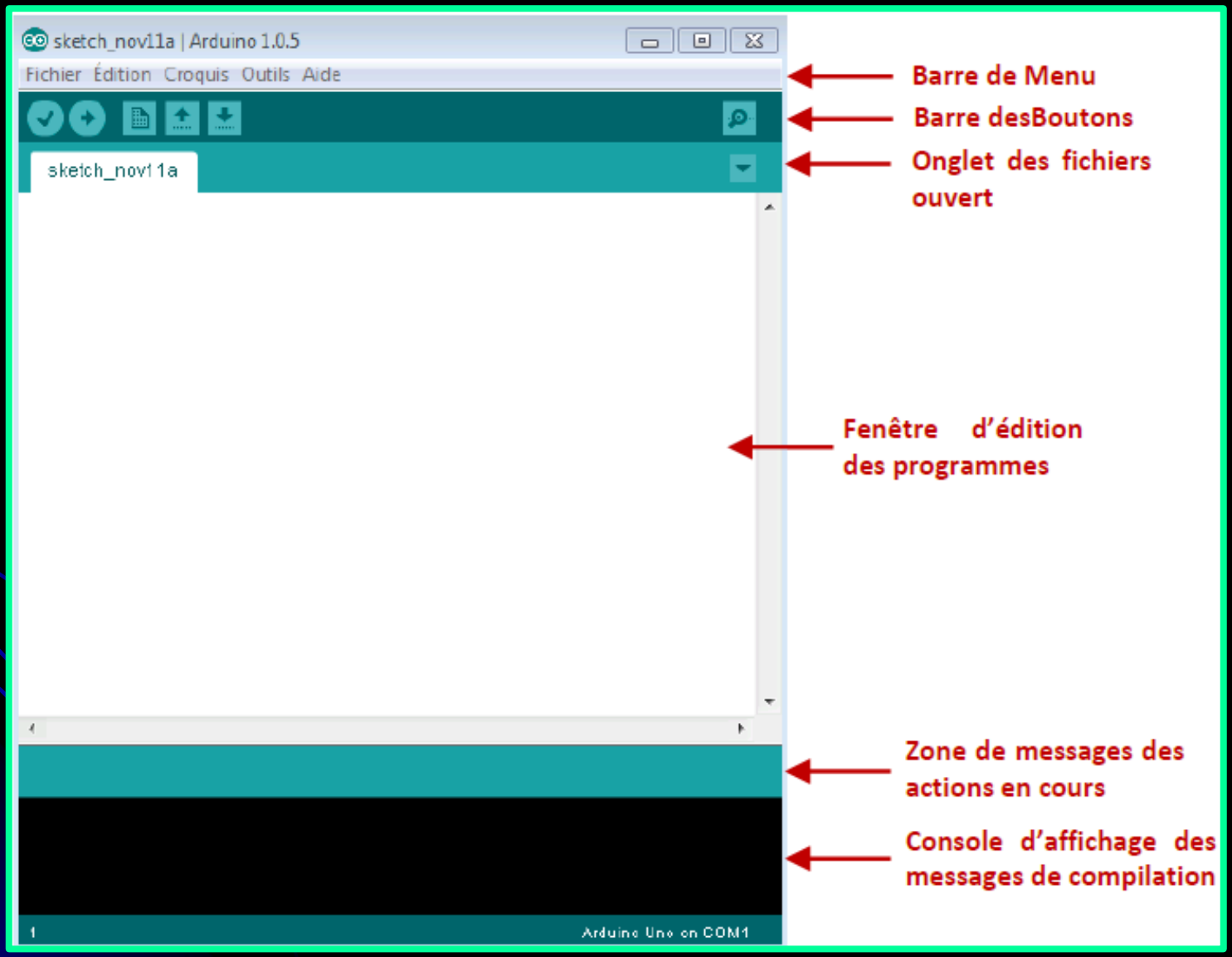
Le logiciel Arduino IDE fonctionne sur Mac, Windows et Linux. C'est grâce à ce logiciel que nous allons créer, tester et envoyer les programmes sur l'Arduino
Le langage de programmation utilisé est un **mélange de C et de C++**, restreint et adapté aux possibilités de la carte

L'IDE est téléchargeable à l'adresse suivante:
http://arduino.cc.

Une fois l'installation proprement effectuée, il est possible maintenant de lancer l'application arduino, en double-cliquant sur le raccourci de l'application.

Le logiciel Arduino IDE

Une fois l'application IDE lancé , on aura la fenêtre suivante:



Le logiciel Arduino IDE

Description de la barre d'outils (des boutons) de l'IDE

- **Vérifier** : Permet de vérifier (ou compiler) le programme avant de l'envoyer sur la carte.
- **Téléverser** : Pour stocker le programme binaire sur la carte et l'exécuter.
- **Nouveau**: Pour créer un nouveau programme.
- **Ouvrir** : Pour ouvrir un programme existant
- **Enregistrer** : Pour sauvegarder le programme
- **Visualiser la console** (Moniteur serie): Ouvrir une fenêtre qui affiche la communication serie entre la carte et le PC.

Le logiciel Arduino IDE

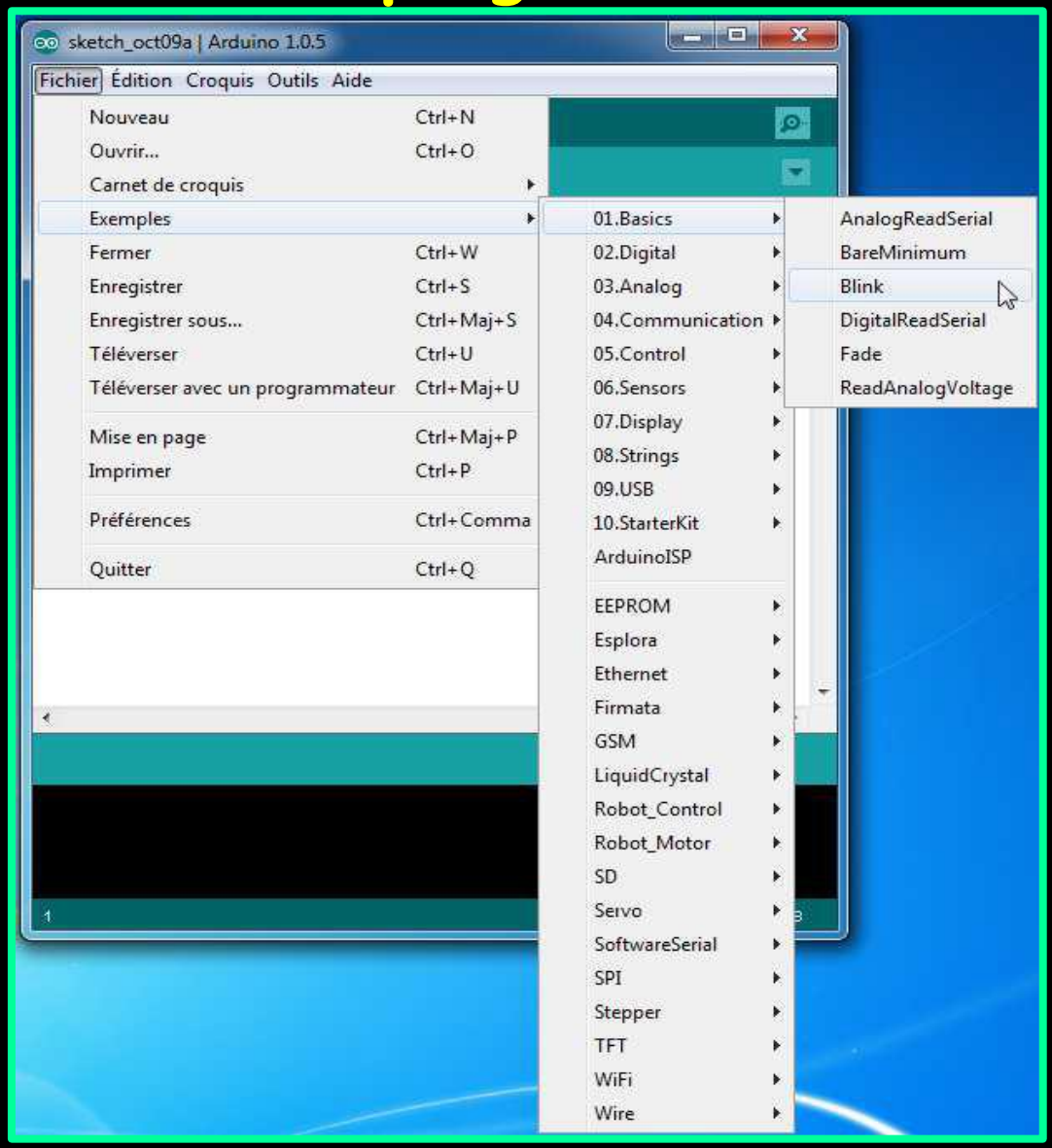
Tester son matériel

Avant de commencer a programmer , il faut, avant toutes choses, tester le bon fonctionnement de la carte. Car ce serait idiot de programmer la carte et chercher les erreurs dans le programme alors que le problème vient de la carte ! Nous allons tester notre matériel en chargeant un programme exemple qui est enregistrer par défaut dans le logiciel ardouino.

Tester la carte Arduino Uno

1ère étape : Ouvrir un programme

Nous allons choisir un exemple tout simple qui consiste a faire clignoter une LED. Son nom est Blink et vous le trouverez dans la categorie Basics :



Le logiciel Arduino IDE

1ère étape : Ouvrir un programme

Une fois que vous avez clique sur Blink, une nouvelle fenêtre va apparaitre. Elle va contenir le programme Blink. Vous pouvez fermer l'ancienne fenêtre qui va ne nous servir plus a rien.

Le logiciel Arduino IDE

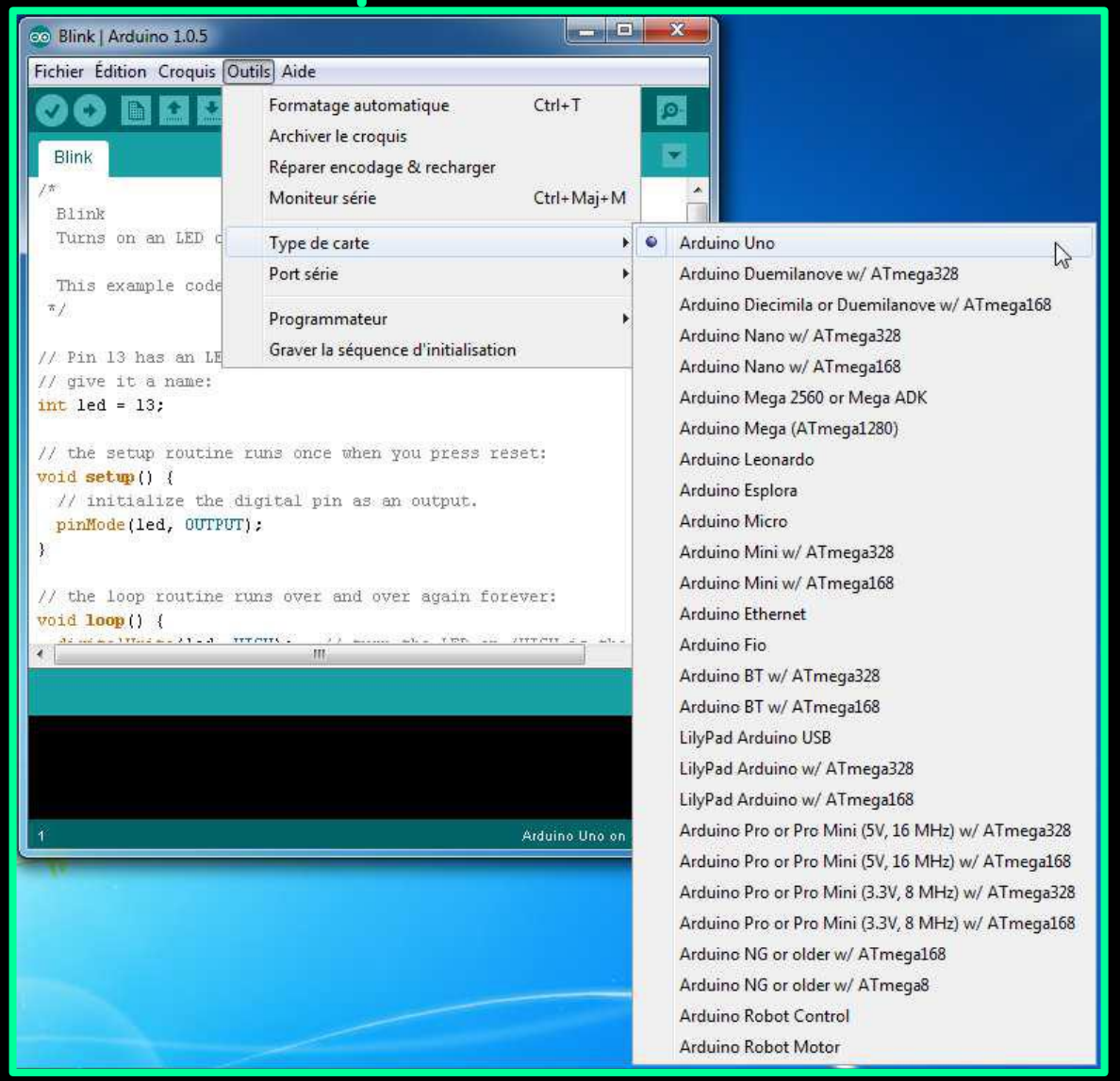
2ieme étape

Avant d'envoyer le programme **Blink** vers la carte, il faut dire au logiciel quel est le **nom de la carte** et sur quel port elle est **branchée**.

Le logiciel Arduino IDE

2ieme étape

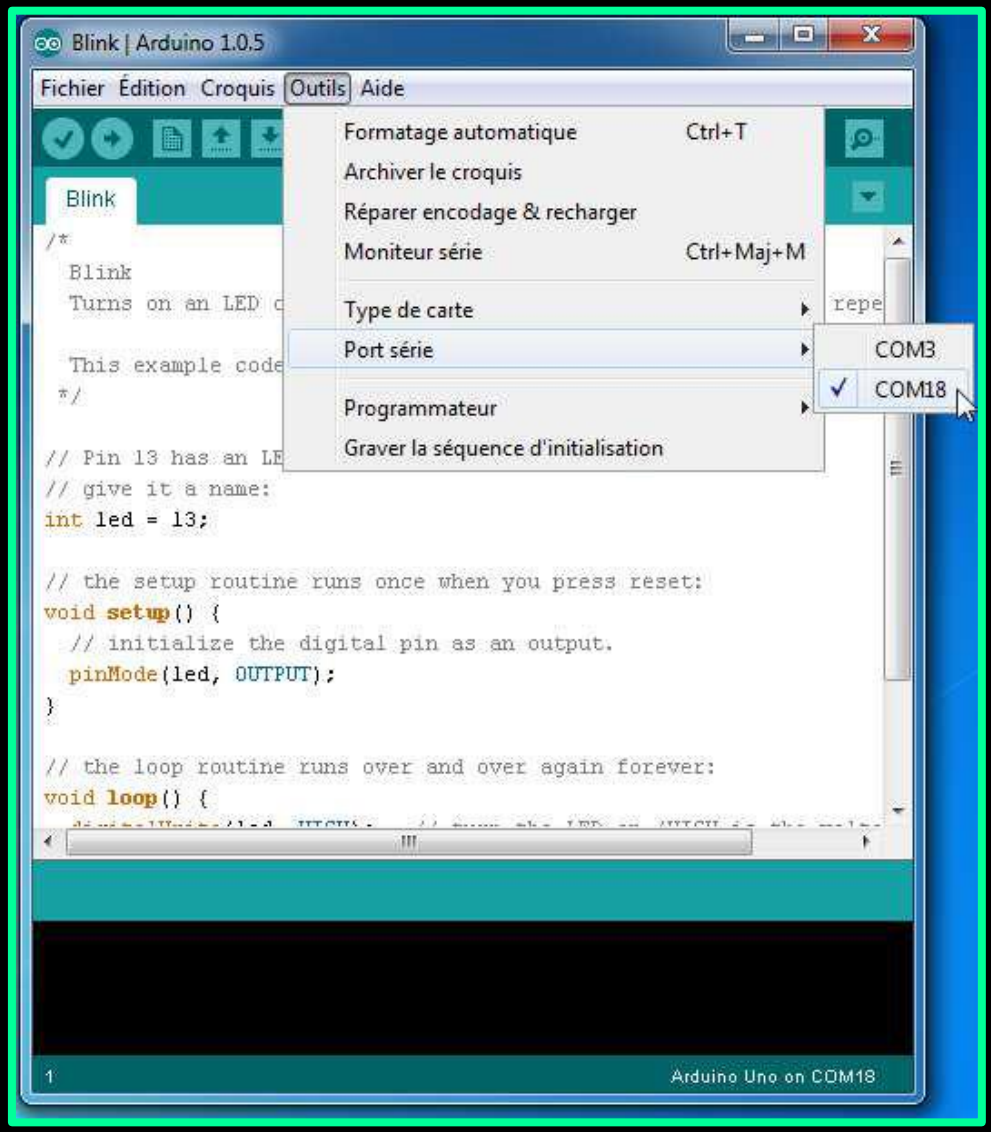
Allez dans le menu " **Tools**" ("**outils**" en francais) puis dans " **Board**" ("**carte**" en francais). Verifiez que c'est bien le nom "**Arduin Uno**" qui est coche. Si ce n'est pas le cas, cochez-le.



Le logiciel Arduino IDE

2ieme étape

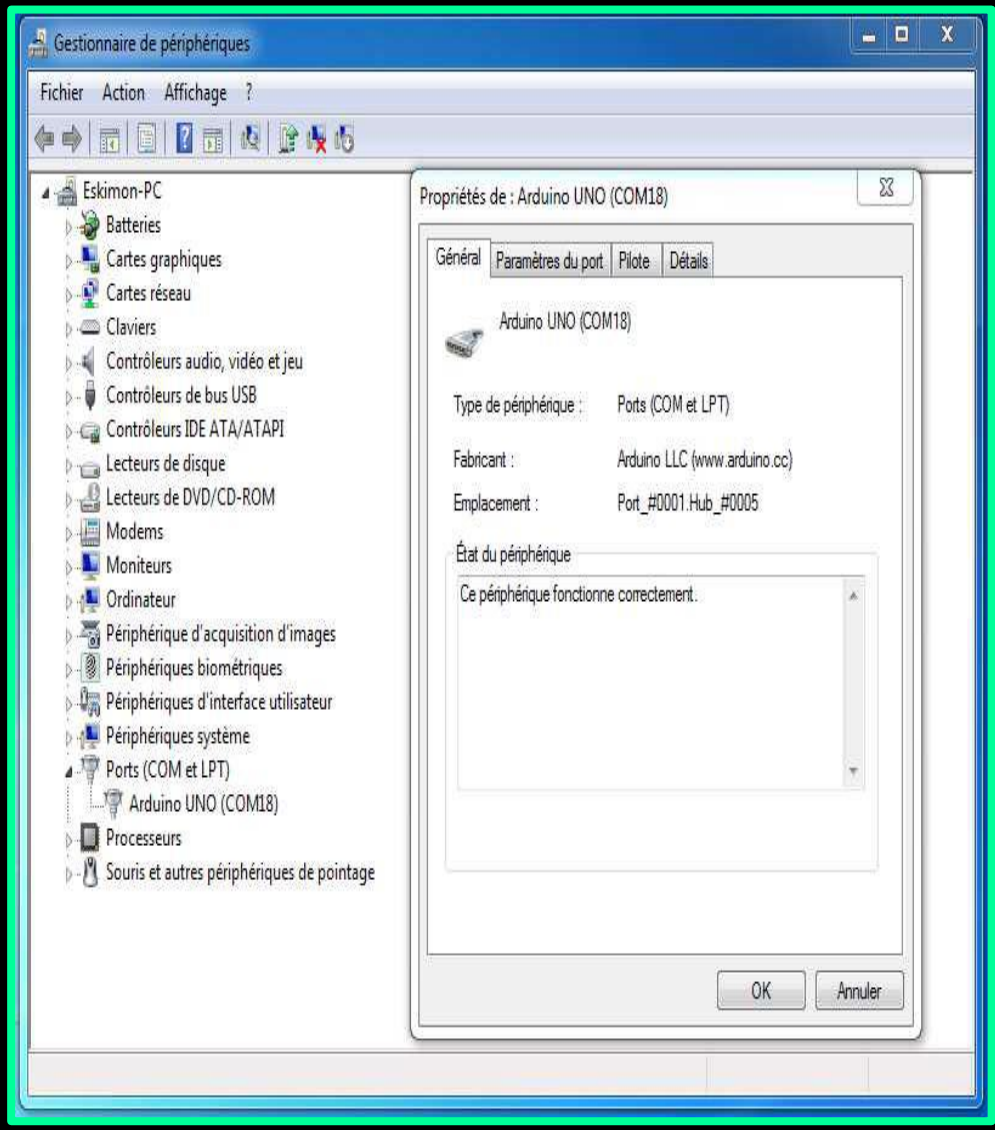
Choisissez le port de connexion de la carte. Allez dans le **menu Tools**, puis **Serial port**. La, vous choisissez le port **COMX**, X etant le numero du port qui est affiche. Ne choisissez pas COM1 car il n'est quasiment jamais connecte a la carte. Dans mon cas, il s'agit de COM5 :



Le logiciel Arduino IDE

2ieme étape

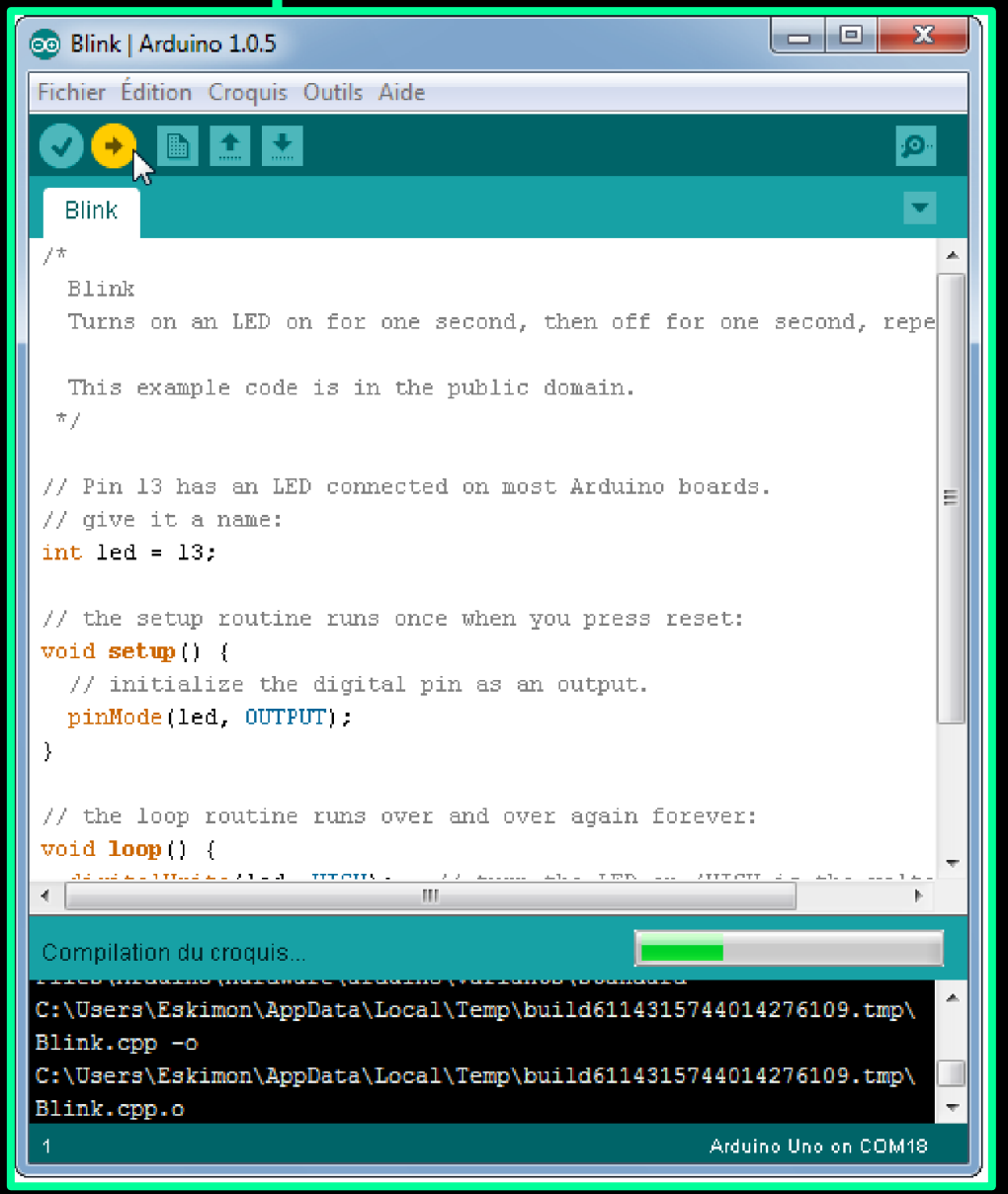
Pour trouver le port de connexion de la carte, vous pouvez aller dans le **gestionnaire de périphérique** qui se trouve dans le panneau de configuration. Regardez a la ligne Ports (**COM et LPT**) et la, vous devriez avoir Arduino Uno (COMX).



Le logiciel Arduino IDE

Dernière étape

Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton **Téléverser**, en jaune-orange sur la photo :



Le logiciel Arduino IDE

Dernière étape

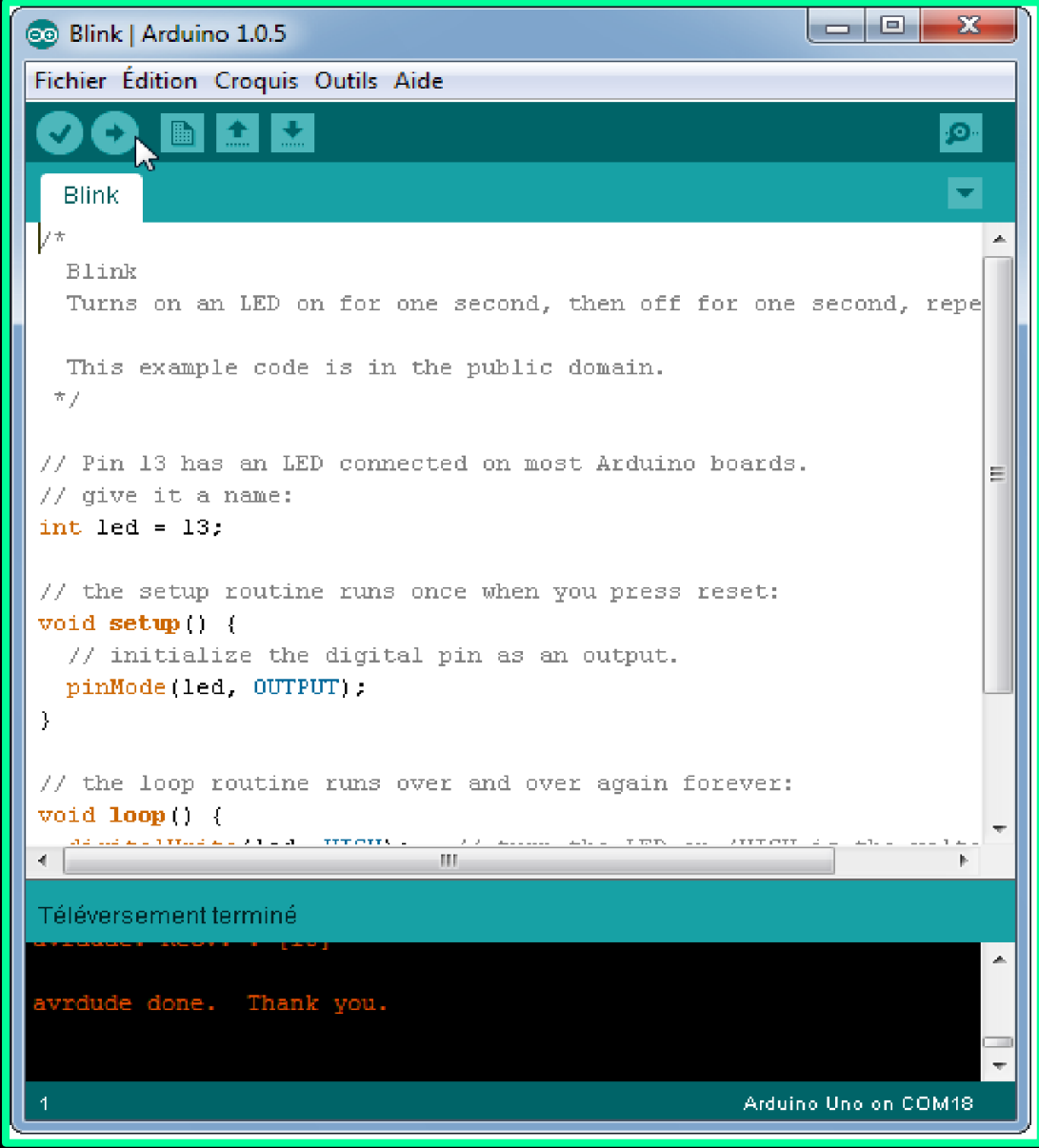
Vous verrez tout d'abord le message "**Compilation du croquis en cours**" pour vous informer que le programme est en train d'étre compile en
langage machine avant d'etre envoyé. Ensuite vous aurez ceci :

En bas dans l'image, vous voyez le texte : "**Téléversement**", cela signifie que le logiciel est en train d'envoyer le programme dans la carte

Le logiciel Arduino IDE

Dernière étape

Une fois qu'il a fini, le programme affiche "**Téléversement terminé**" signale que le programme a bien été charge dans la carte. Si votre materiel fonctionne, vous devriez avoir une LED sur la carte qui clignote :



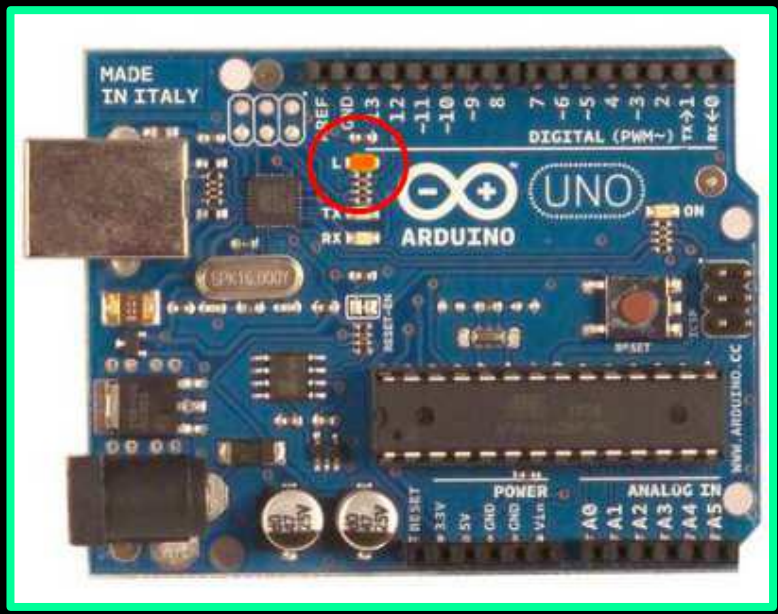
Le logiciel Arduino IDE

REMARQUE

Si vous n'obtenez pas ce message mais plutot un truc en rouge, pas d'inquiétude , le matériel n'est pas forcement défectueux !

En effet, plusieurs erreurs sont possibles:

- l'IDE recompile avant d'envoyer le code, vérifier la présence d'erreur
- La voie série est peut-être mal choisi, vérifier les branchements et le choix de la voie série
- l'IDE est code en JAVA, il peut-être capricieux et bugger de temps en temps (surtout avec la voie serie) : réessayez l'envoi !



Créer votre premier programme Arduino : Blink

Faire clignoter une LED sur la broche 13

```
/*
Code 1 - Edurobot.ch, destiné à l'Arduino
Objectif: faire clignoter la LED montée sur la broche 13
*/
//***** FONCTION SETUP = Code d'initialisation *****
// La fonction setup() est exécutée en premier et une seule fois, au démarrage du programme
void setup() // début de la fonction setup()
{
pinMode(13, OUTPUT); // Initialise la broche 13 comme sortie
Serial.begin(9600); // Ouvre le port série à 9600 bauds
} // fin de la fonction setup()
//***** FONCTION LOOP = Boucle sans fin = coeur du programme *****
// la fonction loop() s'exécute sans fin en boucle aussi longtemps que l'Arduino est sous tension
void loop() // début de la fonction loop()
{
digitalWrite(13, HIGH); // Met la broche 13 au niveau haut = allume la LED
delay(500); // Pause de 500ms
digitalWrite(13, LOW); // Met la broche 13 au niveau bas = éteint la LED
delay(500); // Pause 500ms
} // fin de la fonction loop()
```