



# I. Introduction à l'Arduino

(AOUITI A.Karim et TO. Ala'Eddinne, CSE ENSIT)

## I.1. Description

Les **Arduino** sont des plateformes **soft/hard open source**, c'est à dire, ils sont soumis à la **licence CC (communauté créative)** ce qui exige que le produit est libre à être modifié ou publié gratuitement sans droits d'auteur.

## I.2. Le logiciel ou interface de programmation

Arduino Soft, l'**IDE** ou aussi l'interface de programmation (téléchargeable par Internet), elle est programmée en **Java** et le code source est disponible et gratuit pour ceux qui veulent découvrir les techniques de compilation chez ces **IDE**. Le programmeur soft (**IDE**) utilise un langage de programmation **C/C++ "light"** c'est à dire simple à manipuler dans la programmation des microcontrôleurs.

## I.3. La carte Arduino

C'est une **carte électronique** équipée d'un **microcontrôleur de type AVR** (c'est le nom donné par **Atmel** pour les microcontrôleurs d'architecture **Harvard "AV"** des **RISC "R"** : un RISC = un microcontrôleur, de 8bits = le bus et le processeur à l'intérieur du  $\mu$ contrôleur est de 8bits). Comme les PIC chez **MicroChip**, l'AVR n'est pas d'open source mais l'Arduino, c'est-à-dire, le circuit électronique est open source, en effet chacun peut faire sa propre Arduino et son propre design.

## I.4. La communication avec PC

- [Arduino UNO \(R3 = révision 3\) ou plus](#)

Les AVR de moyenne gamme ne sont pas apte de communiquer via USB (uniquement les  $\mu$ contrôleur hautes performances). Ces cartes utilisent **deux AVR**, un pour le travail, c'est-à-dire, le microcontrôleur que vous allez programmer, l'autre est pour permettre à la carte de communiquer via USB, c'est exactement comme les STM32.

- [Les versions UNO 2009 \(Duemilanove\)](#)

Ces cartes utilisent un convertisseur logique **RS232/USB** appelés **FT232** au lieu du 2ème AVR. Pour cela la programmation est de type **boot loader (série)** et non pas comme le **PICKit2** ou **PICKit3** (chez les PIC) qui utilise une connexion **USB (vrais USB)** ; La communication ici se fait

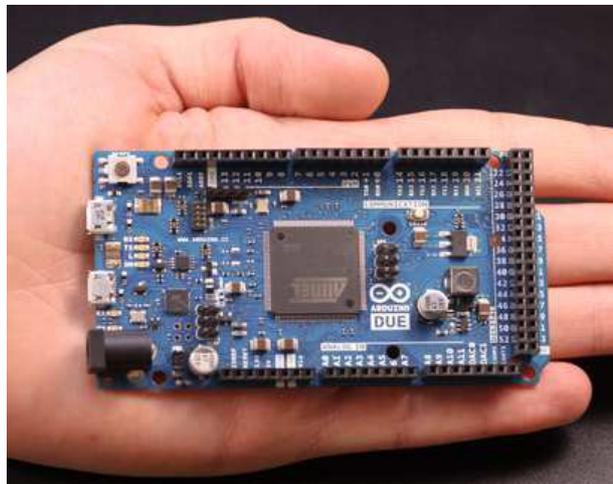
via un moniteur série. Cependant il y a des microcontrôleurs AVR comme **ATMega2560**, **nano Arduino** et **Léonardo** qui utilise des **USB pure**.

### I.5. Les avantages par rapport aux PIC

- **Pas besoin d'un programmeur**, la carte contient déjà un boot loader. Une connexion pour un programmeur mkII dans le cas où l'utilisateur veut programmer via un programmeur Atmel.
- **Simple à programmer**, vous pouvez toujours utiliser les exemples existant dans l'IDE.
- Pas besoin de faire une carte de développement, tout est **prêt à être utilisée**.
- **La flexibilité**: vous pouvez toujours changer l'AVR de la carte Arduino.

### I.6. Les Inconvénients

La programmation n'est pas de très grandes précisions, comme la gestion de temps, une instruction à exécuter peut prendre beaucoup de temps par rapport le vrais temps nécessaire à l'exécution, c'est à dire, une instruction d'un délai d'exécution de **4µs** avec un **AVR**, peut prendre un délai de **80µs** en utilisant une Arduino. Généralement, la gestion de temps chez les Arduino se fait en "ms" et rarement avec des "µs".



*Liens important :*

Télécharger Arduino IDE : <http://arduino.cc/en/Main/Software>