

Chapitre 1 : Introduction aux Systèmes Arduino

I. Introduction

Arduino désigne une carte électronique basé sur un microcontrôleur et doté d'une prise USB permettant de connecter le circuit à un ordinateur, ainsi que plusieurs connexions destinées à relier des composants électroniques externes tels que des moteurs, des relais, des capteurs de lumière, des diodes LED, des microphones, etc.

La carte Arduino peut-être contrôlée depuis un ordinateur ou un programme réalisé sur l'ordinateur puis téléchargé sur la carte qu'on peut ensuite déconnecter pour qu'il fonctionne de façon indépendante.

Les cartes Arduino peuvent être complétés par des cartes d'extension appelées shields qui se branchent pour la plupart directement sur le circuit Arduino.

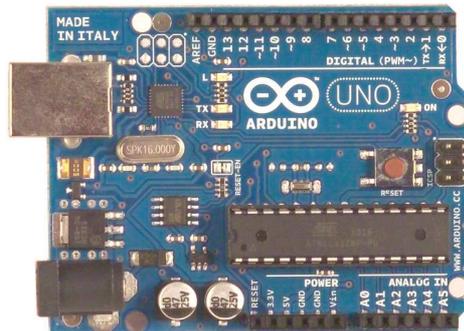


Fig. 1. Arduino UNO

La carte Arduino repose sur un circuit intégré (un mini ordinateur appelé également micro-contrôleur) associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :

- Côté entrées, des capteurs qui collectent des informations sur leur environnement comme la variation de température via une sonde thermique, le mouvement via un détecteur de présence ou un accéléromètre, le contact via un bouton-poussoir, etc.
- Côté sorties, des actionneurs qui agissent sur le monde physique telle une petite lampe qui produit de la lumière, un moteur qui actionne un bras articulé, etc.

II. Caractéristiques

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle dispose de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)), de 6 entrées analogiques, un oscillateur 16Mhz, une connexion USB, une prise de courant, un en-tête ICSP, et un bouton de réinitialisation.

Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont 2 KB sont utilisés par le boot loader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

II.1 Brochage de la carte Arduino

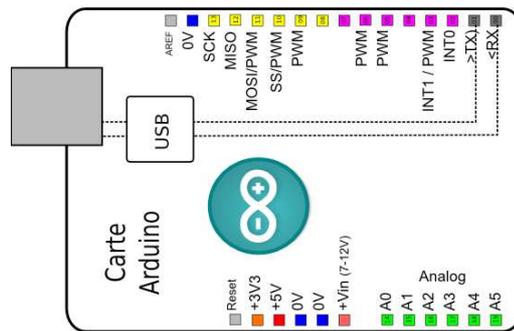


Fig. 2. Brochage de la carte Arduino

II.2. Alimentation

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

Les broches d'alimentation sont les suivantes :

- VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- GND. Broche de masse (ou 0V).

II.3. Mémoire

L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisés par le bootloader). L'ATmega 328 a également 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

II.4. Entrées et sorties numériques

Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Serie:** Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée):** Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique).
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` .
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

II.5. Broches analogiques

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques : elles sont numérotées en tant que broches numériques de 14 à 19.

II.6. Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.
- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

III. Programmation

La carte Arduino Uno peut être programmée avec le logiciel Arduino. Il suffit de sélectionner "Arduino Uno" dans le menu **Tools > Board** (en fonction du microcontrôleur présent sur votre carte). Pour plus de détails sur le langage Arduino, voir la référence du langage Arduino et pour apprendre à programmer en langage Arduino voir la page Apprendre.

Le microcontrôleur ATmega328 présent sur la carte Arduino Uno est livré avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce boot loader communique avec le microcontrôleur en utilisant le Protocol original STK500 (référence, fichiers C).

Vous pouvez bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit").

III.1. Description Le Logiciel Arduino

Le logiciel Arduino a pour fonctions principales :

- de pouvoir écrire et compiler des programmes pour la carte Arduino
- de se connecter avec la carte Arduino pour y transférer les programmes
- de communiquer avec la carte Arduino

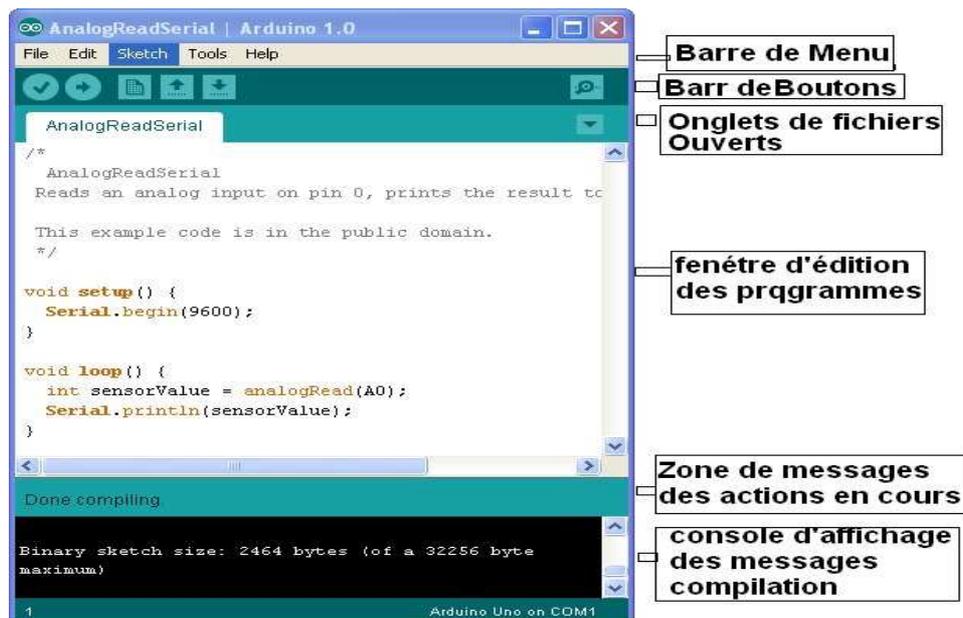


Fig. 3. Fenêtre description de Le logiciel Arduino 1.0

Le logiciel Arduino intègre également :

Un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.

III.2. Principe général d'utilisation

Le code écrit avec le logiciel Arduino est appelé un programme (ou une séquence - sketch en anglais) :

- Ces programmes sont écrits dans l'éditeur de texte. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.
- la zone de messages donne l'état de l'opération en cours lors des sauvegardes, des exportations et affiche également les erreurs.
- La console texte affiche les messages produits par le logiciel Arduino incluant des messages d'erreur détaillés et autres informations utiles.
- la barre de boutons vous permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver votre code, et ouvrir le moniteur série.
- la barre des menus vous permet d'accéder à toutes les fonctionnalités du logiciel Arduino.

III.3. Structure d'un programme Arduino

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

Il y a trois phases principales consécutives:

- La définition des constantes et des variables
- La configuration des entrées et sorties void setup()
- La programmation des interactions et comportements void loop()
- **Les commentaires**

Les premières lignes du programme Blink sont des commentaires (Les commentaires sont des lignes de texte incluses dans le programme et qui ont pour but de vous informer vous-même ou les autres de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur, et ne sont pas programmées dans le microcontrôleur, et par conséquent elles ne n'occupent aucun espace mémoire dans le microcontrôleur ATmega.)

Tout ce qui est entre /* et */ (signes de début et de fin d'un commentaire sur plusieurs lignes) est ignoré par le compilateur Arduino quand il compile le programme (autrement dit, les commentaires ne sont pas programmés dans la carte Arduino). Noter que le * en début de chaque ligne est là uniquement pour faire joli et n'a pas de signification particulière). Les commentaires du début de programme sont là pour les personnes qui liront le code : pour expliquer ce que le programme fait, comment il fonctionne ou pourquoi il est écrit de cette façon. C'est une bonne habitude de commenter (abondamment...) vos programmes, et de mettre à jour vos commentaires quand vous modifiez le code. Cela aidera les autres (et parfois vous-même..) pour apprendre à partir de votre code ou le modifier.

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette boucle se déroule des milliers de fois par seconde et anime la carte.

IV. Conclusion

Le système Arduino est une plateforme puissante, flexible et transparente ; grâce à la simplicité d'utilisation et de programmation ; ainsi que le principe de source ouvert permet l'échanger d'idées entre développeurs, enrichir la bibliothèque des codes sources et faciliter la gestion des périphériques, divers montages peuvent être réalisés avec la moindre des composants extérieurs. Le logiciel Arduino est un véritable environnement intégré de développement dédié au langage Arduino. Le logiciel Arduino nous permettra d'écrire nos programmes, de les déboguer, de les compiler et de les transférer dans la carte Arduino