

Objectif : L'élève doit être capable d'identifier la structure fonctionnelle d'un microcontrôleur et les outils de programmation.

LA CARTE ARDUINO UNO

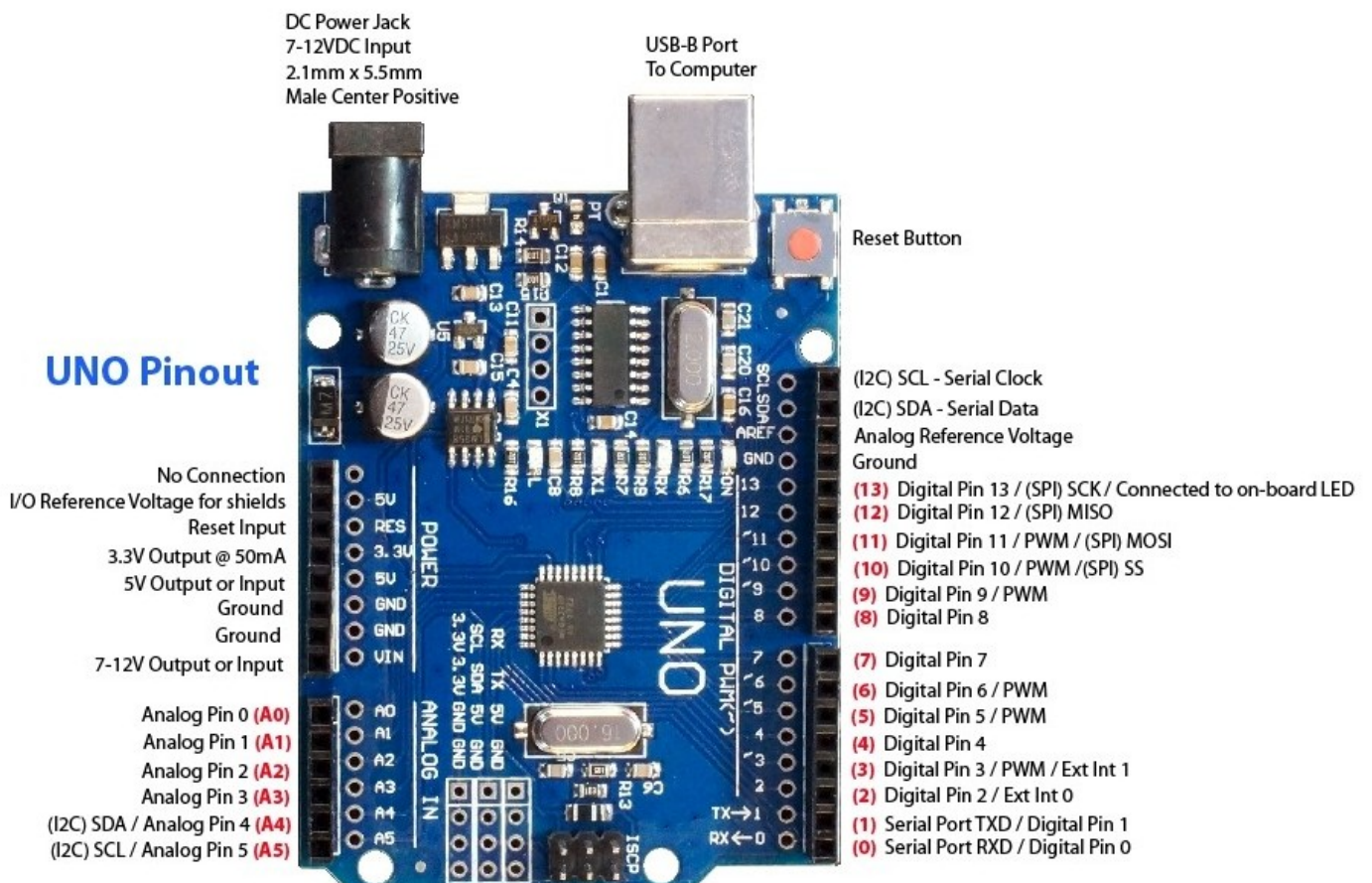
1 – Présentation

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle dispose :

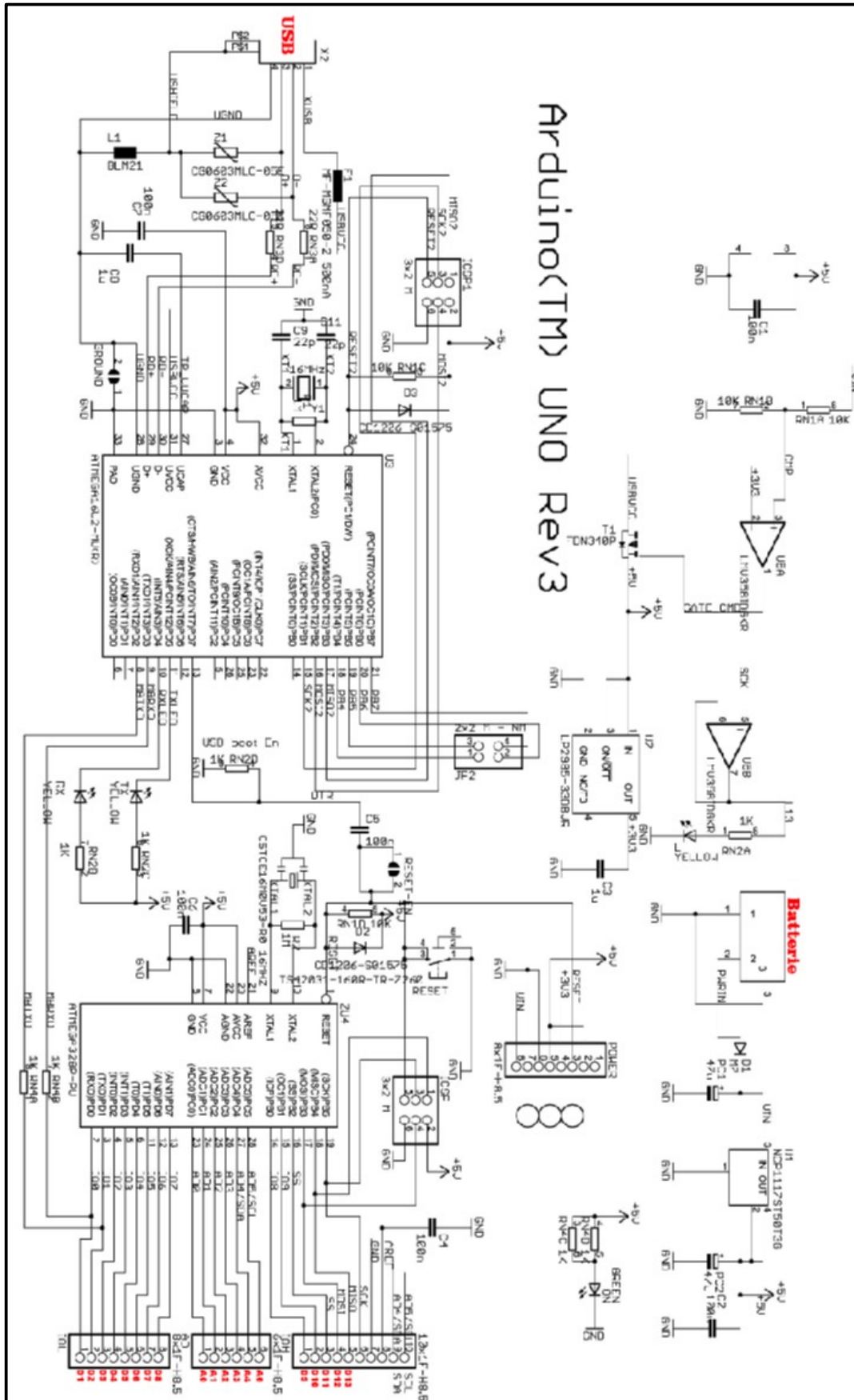
- de 14 broches numériques d'entrées/sorties,
- de 6 entrées analogiques,
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP,
- d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

2 – Brochage



3 – Schéma de principe



4 – L'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.

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.

- 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 (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). 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 toute 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).

5 – 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'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

De plus, certaines broches ont des fonctions spécialisées :

- Communication Série: 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) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- 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 (ou TWI - Two-Wire interface - interface "2 fils") .
- 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.

6 – 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)`.

7 – 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.
- 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 (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). 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 toute 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).

8 – Communication

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier .inf est requis.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

- Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails.
- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

9 – Programmation

La carte Arduino Uno peut être programmée avec le logiciel Arduino (à télécharger ici). 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 bootloader communique avec le microcontrôleur en utilisant le protocole original STK500 (reference, 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" en français); voir ces instructions (en anglais) pour plus de détails.

Le source du code pour le circuit intégré ATmega8U2 est disponible. L'ATmega8U2 est chargé avec un bootloader DFU qui peut être activé en connectant le cavalier au dos de la carte (près de la carte de l'Italie) et en réinitialisant le 8U2. Vous pouvez alors utiliser le logiciel FLIP de chez Atmel (Windows) ou le programmeur DFU (Mac OS X et Linux) pour charger le nouveau code. Ou bien vous pouvez utiliser le connecteur ICSP avec un programmeur externe (pour réécrire le bootloader DFU).

10 – Réinitialisation (logicielle) automatique

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino UNO a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel tournant sur l'ordinateur. Une des broches matérielles de contrôle du flux (DTR) du circuit intégré ATmega8U2 est connecté à la ligne de réinitialisation de l'ATmega 328 via un condensateur de 100 nanofarads. Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel Arduino utilise cette possibilité pour vous permettre de transférer votre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications. Quand la carte Uno est connectée à un ordinateur fonctionnant soit sous Mac Os X ou Linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel (via le port USB). Pendant la demi-seconde suivante, le bootloader s'exécute sur la carte Uno. Puisqu'il est programmé pour ignorer les données mal formatées (càd toute donnée en plus du transfert d'un nouveau programme), il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme s'exécutant sur la carte reçoit une configuration initiale ou d'autres données quand il démarre, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte Uno comporte une piste (du circuit imprimé) que vous pouvez couper pour désactiver la réinitialisation automatique. Les bords de chaque côté de la piste peuvent solidariser ensemble pour la réactiver. Vous pouvez également désactiver la réinitialisation automatique en connectant une résistance de 110 Ohms entre le +5V et la ligne de réinitialisation (reset); voir ce fil du forum (en anglais) pour plus de détails.

Protection du port USB contre la surcharge en intensité

11 – Protection du port USB contre la surcharge en intensité

La carte Arduino Uno intègre un polyfusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

12 – Caractéristiques Mécaniques

Les longueurs et largeurs maximales de la Uno sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.