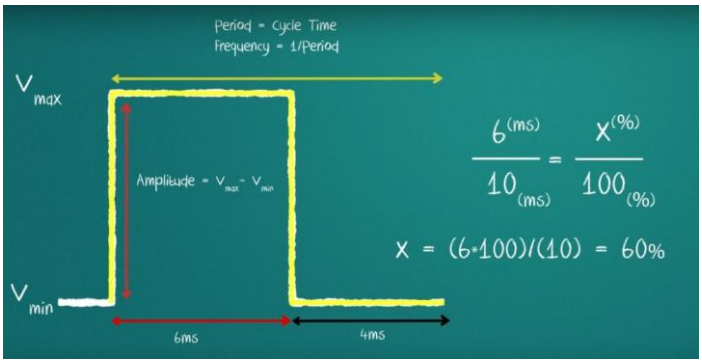
1. Qu'est-ce que la modulation de largeur d'impulsion ?

1.1. Introduction au PWM

PWM signifie **Pulse Width Modulation** (modulation de largeur d'impulsion). D'après son nom, il est clair que dans cette technique, la largeur des impulsions d'une forme d'onde est contrôlable (change). Cela signifie que la durée pendant laquelle une impulsion se trouve à l'état **HAUT** ou à l'état **BAS** à partir d'un moment donné est contrôlable.

Dans cette technique, nous utilisons des méthodes numériques pour obtenir un résultat analogique. La commande numérique est utilisée pour créer une forme d'onde carrée en activant et en désactivant un signal de manière répétée. Ce schéma ON et OFF génère une tension VCC complète à la position ON et 0V à la position OFF.

Le rapport cyclique est défini comme la durée moyenne de mise en marche d'une impulsion divisée par la durée TOTALE de l'impulsion, de sorte qu'en principe, le rapport cyclique des impulsions est modifié dans la modulation de largeur d'impulsion.



La formule du cycle d'utilisation est donc présentée dans l'expression suivante :

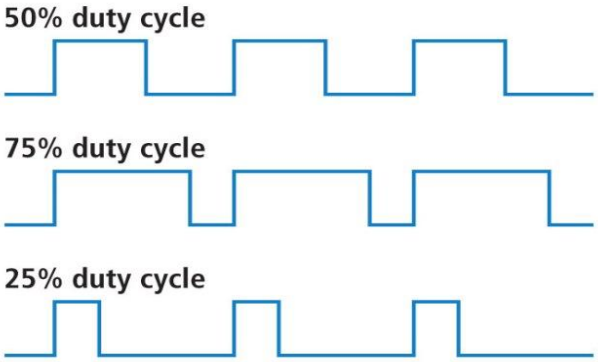
Duty Cycle = (On time of signal / total timer period of signal)

1.2. Comment calculer la tension de sortie du signal PWM ?

Voyons maintenant comment calculer un cycle de fonctionnement. Pour calculer le rapport cyclique, vous devez connaître la durée de la période pendant laquelle le signal est élevé. Réglons notre temps haut à 6 millisecondes et notre temps bas à 4 millisecondes. La période totale est de 10 ms. Utilisons maintenant une règle simple pour calculer le pourcentage de la période pendant laquelle le signal est élevé par rapport à la période totale.

Duty cycle = 6ms / 10ms = 0.6

En résolvant ce problème, nous obtenons un cycle d'utilisation de 0,6 qui est une quantité sans unité. Nous mesurons toujours le cycle d'utilisation en pourcentage. Le rapport cyclique maximal peut être de 1 ou de 100 % lorsque l'heure ou le temps fort du signal est égal à la période totale du signal. De même, le rapport cyclique minimum sera de 0 ou 0 % lorsque le signal est désactivé pendant toute la durée de la minuterie. Vous pouvez voir ci-dessous une image des signaux avec différents rapports cycliques.



1.3. Amplitude du PWM

L'amplitude de la modulation de largeur d'impulsion est un autre concept important à discuter pour bien comprendre cette notion. L'amplitude est la différence entre une tension maximale et une tension minimale du signal.

Amplitude = Vmax – Vmin

1.4. Techniques de génération de modulation de largeur d'impulsion

Deux méthodes sont utilisées pour générer la PWM :

- La génération de PWM en utilisant des circuits numériques comme le microcontrôleur.
- La génération de PMW à l'aide de circuits analogiques tels que les amplificateurs opérationnels et les circuits comparateurs.

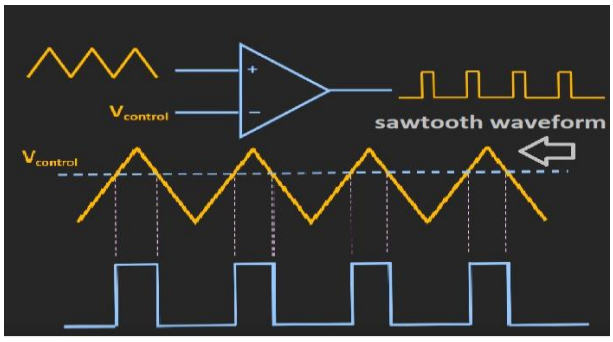
1.5. Génération de modulation de largeur d'impulsion à l'aide d'un microcontrôleur

Pour générer un signal numérique avec un rapport cyclique variable, il est toujours recommandé d'utiliser un microcontrôleur comme Arduino. En effet, ces microcontrôleurs

disposent d'un module intégré pour générer des signaux numériques. Vous pouvez facilement régler le rapport cyclique à l'aide de la programmation du microcontrôleur. Consultez ces articles pour générer des PWM à l'aide d'un microcontrôleur.

1.6. Génération de modulation de largeur d'impulsion à l'aide de circuits analogiques

La façon la plus simple de générer un signal PWM est d'utiliser un amplificateur opérationnel. Pour générer un signal numérique avec un amplificateur opérationnel, nous utilisons un amplificateur opérationnel comme circuit de comparaison. L'amplificateur opérationnel est constitué de deux bornes : une borne non inverseuse et une borne inverseuse. Nous appliquons une onde triangulaire à l'entrée non inverseuse et une tension de contrôle à la broche inverseuse de l'amplificateur opérationnel. L'image ci-dessous montre l'ensemble des processus de génération du signal pwm en utilisant l'amplificateur opérationnel.



Donc, pour comprendre le fonctionnement de cette méthode, il faut d'abord comprendre le fonctionnement de l'amplificateur opérationnel comme circuit comparateur. Comme vous pouvez le voir sur les photos ci-dessus, lorsque la tension triangulaire appliquée à l'entrée non inverseuse est inférieure à la tension de commande appliquée à la broche inverseuse de l'amplificateur opérationnel, la sortie du circuit comparateur sera faible et lorsque la tension du signal triangulaire est supérieure à la tension de commande, la sortie du comparateur sera élevée. Ainsi, le temps d'activation du signal numérique ou de la modulation de largeur d'impulsion dépend de l'amplitude de la tension de commande. Ainsi, la tension de commande moyenne et le rapport cyclique sont inversement proportionnels l'un à l'autre. Parce que le rapport cyclique est directement proportionnel au temps de fonctionnement du signal PWM. Pour obtenir un rapport cyclique plus élevé, nous devons diminuer la valeur de

la tension de commande. Pour obtenir un rapport cyclique plus faible, nous devons augmenter la valeur de la tension de commande. C'est donc dire à quel point il est facile de générer des PWM à l'aide de composants électroniques analogiques.

1.7. Autre méthode pour générer des PWM

Il existe sur le marché de nombreux circuits intégrés qui sont utilisés pour produire des signaux PWM et qui ont également la capacité de générer des signaux numériques à rapport cyclique variable. Les noms de certains d'entre eux sont donnés ci-dessous :

- **555 timer IC**
- **SG3525 pulse width modulation controller**

1.8. Applications de la modulation de largeur d'impulsion

J'ai mentionné quelques applications au début de cet article. Mais vous pouvez voir mon projet dans lequel j'ai utilisé le PWM pour diverses applications comme les onduleurs, l'onduleur triphasé et l'électronique de puissance.

- Onduleur solaire utilisant le SG3525
- Génération de SPWM à l'aide de pic16f877a
- Onduleur sinusoïdal triphasé utilisant Arduino
- SVPWM triphasé utilisant le microcontrôleur Pic

2. Génération de PWM à l'aide d'Arduino Mega 2560

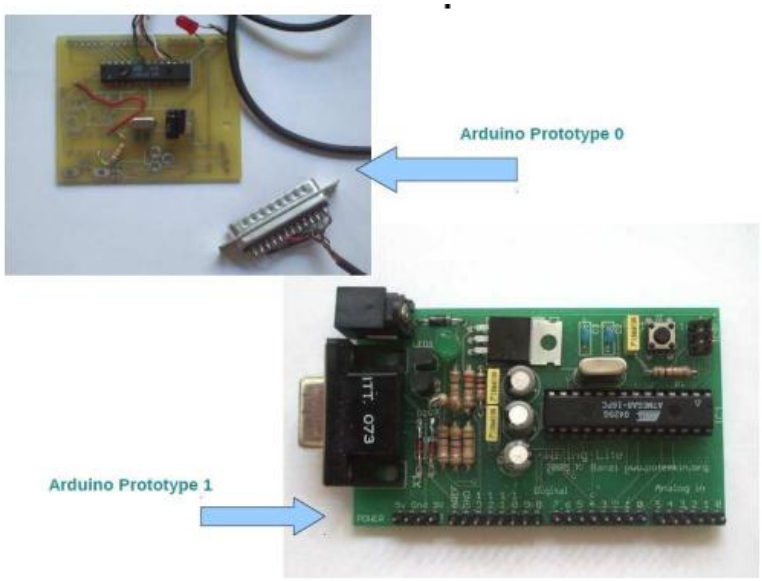
2.1. Vous avez dit arduino ?

- Pont tendu entre le **monde réel** et le **monde numérique, Arduino** permet d'**étendre** les capacités de relations **humain/machine ou environnement/machine.**
- **Arduino** est un projet en **source ouverte (open source)** : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions
- **Arduino** est une **plate-forme de prototypage** d'objets interactifs à usage créatif constituée d'une **carte électronique** et d'un **environnement de programmation**.

  Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur
- de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

2.2. Arduino en résumé

2.3. Historique



2.4. Les créateurs : des artistes au sens premier du terme

2.5. Leur objectif : Processing pour le Hardware !

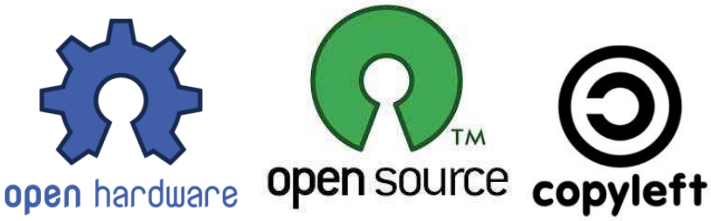
a. Qu'est-ce que Processing ?

- un langage de programmation et un environnement de développement créé par Benjamin Fry et Casey Reas, deux artistes américains.
- particulièrement adapté à la **création plastique et graphique** interactive
- Le logiciel fonctionne sur Macintosh, sous Windows et sous Linux, car il est basé sur la plate-forme Java — il permet d'ailleurs de programmer directement en langage Java.

b. Pourquoi ?

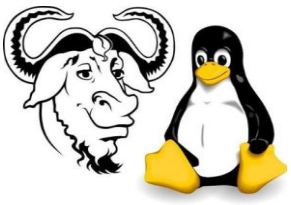
Matériel robotique excessivement cher Le matériel est « open source » :

- On peut le copier, le fabriquer et le modifier librement.



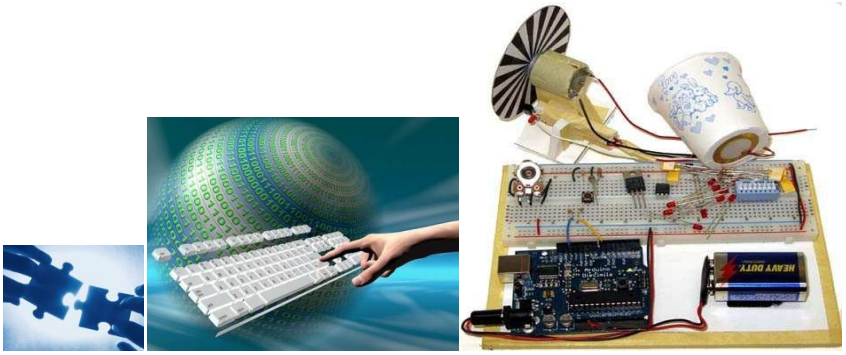
Le logiciel est libre :

- On peut l'utiliser et le modifier librement.



Sur l'Internet, on trouve :

- Une communauté d'utilisateurs.
- Des guides d'utilisation.
- Des exemples.
- Des forums d'entraide.



c. Avantages

- **Pas cher !**
- Environnement de programmation clair et **simple.**
- **Multiplate-forme :** tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.

- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc.)
- Existence de **« shield »** (boucliers en français)

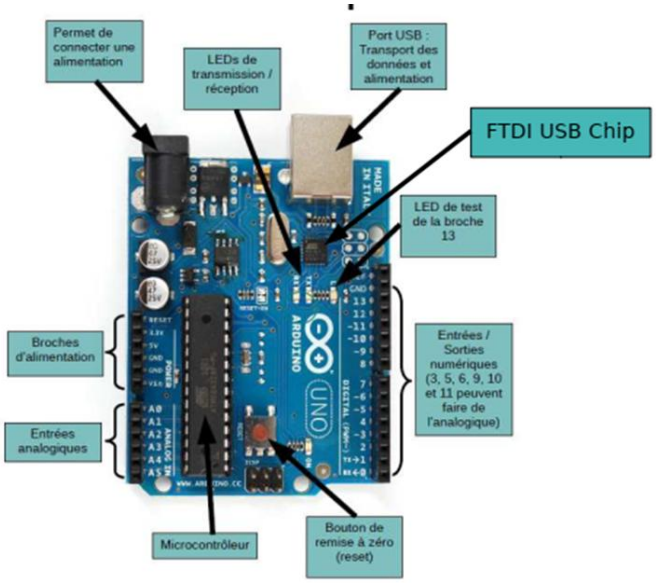
d. C'est quoi « pas cher » ?

- Prix d'une carte Arduino Uno = 25 euros
- Logiciel = 0 euros
- Support et assistance = 0 euros (forums)

e. Domaine d'utilisation

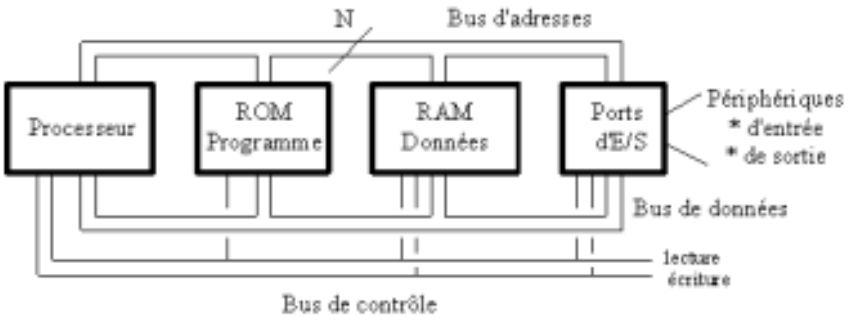
- Physical computing : Au sens large, **construire des systèmes physiques** interactifs qui utilisent **des logiciels et du matériel** pouvant s'interfacer avec **des capteurs et des actionneurs.**
- Électronique industrielle et embarquée
- Art / Spectacle
- Domotique
- Robotique
- Modélisme
- DIY (Do-It-Yourself), Hacker, Prototypage, Education, Etc.

2.6. La carte électronique Arduino



2.6.1. Qu'est-ce qu'un microcontrôleur ?

Un microcontrôleur intègre sur un unique die (circuit intégré) : **un processeur (CP**U), avec une largeur du chemin de données allant de 4 bits pour les modèles les plus basiques à 32 ou 64 bits pour les modèles les plus évolués ; de la **mémoire vive (RAM)** pour stocker les données et variables ; **de la mémoire pour stocker le programme : ROM** (mémoire morte) et/ou EPROM, EEPROM, Flash ; souvent un oscillateur pour le cadencement. Il peut être réalisé avec un quartz, un circuit RC ou encore une PLL1 ;



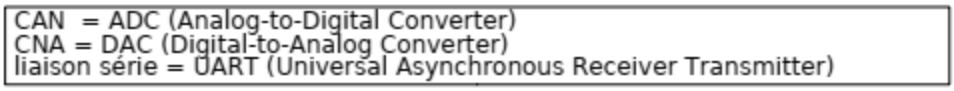
µcontrôleur : circuit intégré qui rassemble les éléments essentiels d'un ordinateur : **processeur, mémoires** (mémoire morte et/ou programmable pour le programme, mémoire vive pour les données), **unités périphériques et interfaces d'entrées sorties**

Ils sont fréquemment **utilisés dans les systèmes embarqués**, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Qu'est-ce qu'un microcontrôleur ? Des périphériques, capables d'effectuer des tâches spécifiques. On peut mentionner entre autres :

- les **convertisseurs analogiques-numériques** (CAN) **(donnent un nombre binaire à partir d'une tension électrique),**
- les **convertisseurs numériques-analogiques** (CNA) (**effectuent l'opération inverse**),
- les **générateurs de signaux à modulation de largeur d'impulsion** (MLI, ou en anglais, **PWM pour Pulse Width Modulation**),
- les timers/compteurs (compteurs d'impulsions d'horloge interne ou d'événements externes),

- les chiens de garde (watchdog),
- les comparateurs (comparent deux tensions électriques),
- les contrôleurs de bus de communication (UART, I^2C, SSP, CAN, FlexRay, USB, Ethernet, etc.).



Les créateurs : des artistes au sens premier du terme

2.7. Arduino :

Sortie en 2005 comme un modeste outil pour les étudiants de Banzi à l'Interaction Design Institute Ivrea (IDII), Arduino a initié une révolution DIY dans l'électronique à l'échelle mondiale.

Arduino est une plateforme électronique open-source basée sur un microcontrôleur ATmega. Elle est très populaire chez les roboticiens, les amateurs et les professionnels. Son succès est principalement dû à sa conception de type ouverte et à sa facilité extrême d'utilisation, tant au niveau du matériel (module) que du logiciel (IDE open-source facile à utiliser). Il est par ailleurs très simple de débuter avec l'Arduino grâce aux très nombreux projets et matériels d'apprentissage facilement accessibles.

2.8. Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques-unes afin d'éclaircir l'évaluation de ce produit scientifique et académique:

Parmi ces types, nous avons choisi une carte Arduino Mega. L'intérêt principal de cette carte est de faciliter la mise en oeuvre d'une telle commande qui sera détaillée par la suite.

2.9. Description générale de la carte Arduino Mega 2560

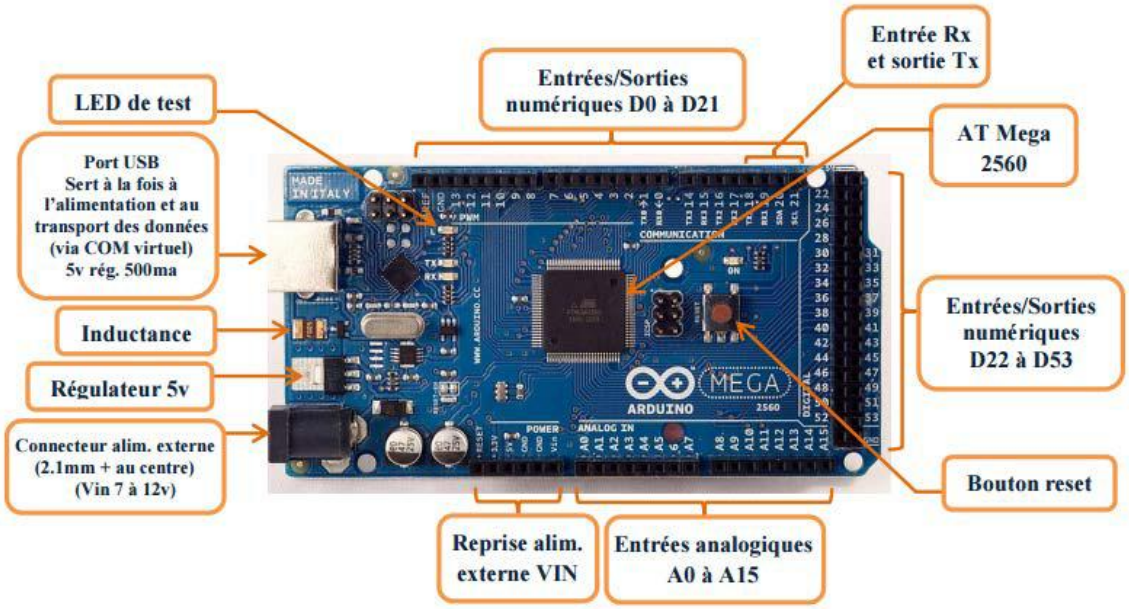


Figure.3.1. Description de la carte Arduino MEGA 2560

2.10. Caractéristiques de l'Arduino Méga 2560 :

Cette carte dispose :

- de 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties
- PWM (largeur d'impulsion modulée) ;
- de 16 entrées analogiques (qui peuvent également être utilisées en broches
- entrées/sorties numériques);
- de 4 UART (port série matériel) ;
- d'un quartz 16Mhz ;
- d'une connexion USB ;
- d'un connecteur d'alimentation jack ;
- d'un connecteur ICSP (programmation "in-circuit") ;
- et 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).

2.11. Caractéristique technique de la carte Arduino Mega 2560

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmateur dédié ne soit pas nécessaire.

2.12. Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

Tableau 3.1: Constitution de la carte Arduino Mega 2560

Microcontrôleur	ATMEGA2560
Tension de fonctionnement	5V
Tension d'alimentation	7 à 12V
Broches E/S numérique	54 (dont 14 disposent de sortie PWM)
Broches d'entrées analogiques	16
Vitesse d'horloge	16 MHz
Mémoire programme Flash	25 6KB dont 8 KB utilisés en bootloader
Mémoire SRAM	8 KB
Mémoire EEPROM	4 KB

a. Le Microcontrôleur ATMega2560

Un microcontrôleur ATMega2560 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit, c'est le processeur de la carte, s'occupe de tout ce qui est calculs, exécution des instructions du programme et gestion des ports d'entrée/sortie.

Microcontrôleur ATMega2560

b. Les mémoires

L'ATmega 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader), également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

c. Les sources de l'alimentation de la carte

La carte Arduino Mega 2560 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 ce dernier. 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).

• **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 ou de tout autre source d'alimentation régulée.

• **3.3V** : 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. L'intensité maximale disponible sur cette broche est de 50mA

• **GND** : Broche de masse (0V).

d. Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Mega peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. 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 digital-Write(broche, HIGH). De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série**: Port Série Serial : 0 (RX) et 1 (TX); Port Série Serial 1: 19 (RX) and 18 (TX); Port Série Serial 2: 17 (RX) and 16 (TX); Port Série Serial 3: 15 (RX) and 14 (TX).
- **Interruptions Externes**: Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4) ,20 (interrupt 3), et 21 (interrupt 2).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.
- **mpulsion PWM** (largeur d'impulsion modulée): Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().
- **SPI** (Interface Série Périphérique): Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique).
- **I2C**: Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C.
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13.

e. Broches analogiques

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analog-Read() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023). Les broches analogiques peuvent être utilisées en tant que broches numériques.

f. 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 analog-Reference.

• **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil.

g. Les Accessoires de la carte Arduino

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

h. Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

- **Le module Arduino Bluetooth**

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion sérielle Bluetooth à la place d'une connexion USB, très faible consommation d'énergie mais aussi très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.

- **Le module shield Arduino Wifi**

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.

- **Le Module XBee**

Ce module permet de faire de la transmission sans fil, faible distance /consommation /débit/ prix.

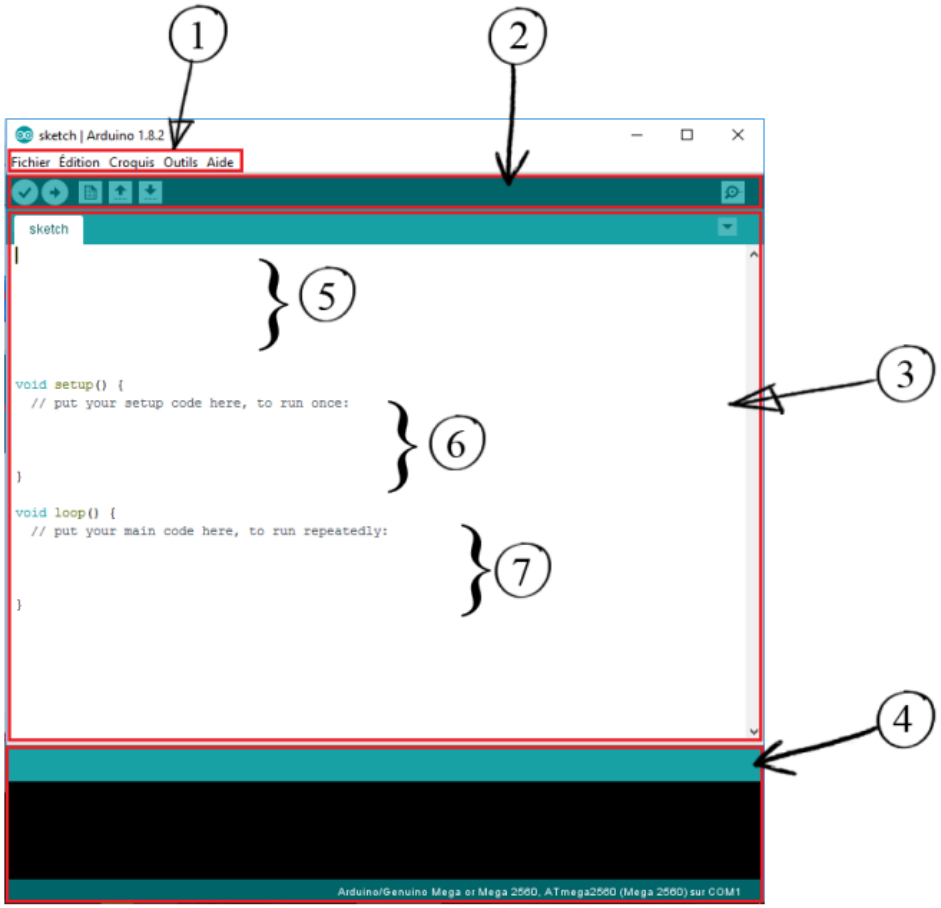
2.13. Partie programme

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

a. L'interface

L'interface du logiciel se présente de la façon suivante :

- options de configuration du logiciel.
- boutons pout la programmation des cartes.
- Zone pour programmer.
- débogueur (affichage des erreurs de programmation).
- partie déclaration de variables (globales).
- Initialisation « Void Setup () {} » : Au démarrage de l'Arduino toutes les instructions comprise entre les deux accolades serrons exécuter qu'une seul fois.
- Boucle principale « Voidloop(){} » : Les instruction sont répéter indéfiniment tant que l'Arduino fonctionne.



b. Les boutons du logiciel

1. Vérifier : permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme.
2. Téléverser : compiler et envoyer le programme vers la carte.
3. Nouveau : créer un nouveau fichier.
4. charger un programme existant.
5. Sauvegarder le programme en cours.
6. Moniteur série : de base sur la carte Arduino on ne peut pas afficher de texte, il faut ajouter un module d'affichage ou bien se servir du moniteur série pour utiliser l'écran de notre ordinateur pour savoir où on en est dans l'exécution du programme

3. L'environnement de la programmation

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino.

3.1. Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

3.2. Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino Mega2560) et le numéro de port USB (COM).

3.3. Description du programme

Un programme 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 codes.

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

3.4. Définition des variables

Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la troisième sortie numérique ; cette variable doit être définie et nommée; la syntaxe est pour désigner un nombre entier est **int.**

3.5. Configuration des entres et des sorties void setup ()

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on configure pin en sortie ; pin mode (nom, état).l'état est soit OUTPUT pour les sorties ou INPUT lorsqu'il s'agit d'entrées.

3.6. Programmation des interactions voidloop

Dans cette boucle, on définit les opérations à effectuer dans l'ordre digitalwrite(nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- Delay (temps en mili-seconde) est la commande d'attente entredeux instructions.
- chaque ligne d'instruction est terminée par un point-virgule.
- ne pas oublier les accolades qui encadrent la boucle

3.7. Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).

On vérifie que notre montage fonctionne.

Méthode expérimentale « Génération de PWM à l'aide d'Arduino Mega 2560 »