Introduction sur ARDUINO

I. Introduction

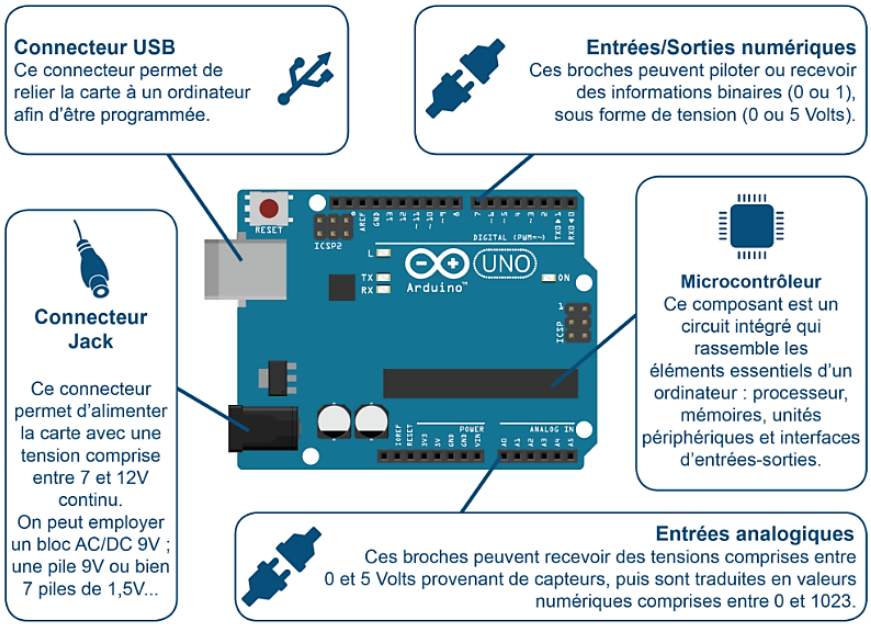
Arduino est une petite carte de prototypage dotée d'un microcontrôleur ATMEGA et une interface USB-Série qui permet de la connecter à un port USB du PC pour téléverser les programmes dans l'ATMEGA. La carte possède des connecteurs permettant d'accéder facilement aux E/S du Microcontrôleur.

Il existe plusieurs types de cartes Arduino et leur choix dépond d'un cahier des charges ainsi que les performances demandées :

UNO, LEONARDO, MICRO, NANO, MENI, DUE, MEGA, YUN …

Dans ce cours tous les programme seront testés dans l'Arduino UNO ;

II. Caractéristiques de l'Arduino UNO



1. Le processeur

L'Arduino est composée d'un processeur ATMEGA328 qui présente 32Ko de mémoire programme (flash), 2ko de mémoire de données (SRAM) et un Horloge 16 Mhz.

2. Les Entrées Sorties :

L'Arduino est constituée de :

- 13 E/S numériques dont 6 peuvent fonctionner comme sorties PWM (MLI)
- 6 entrées analogiques (utilisés pour convertir des grandeurs analogiques au grandeurs numérique par un CAN)
- Un bus de communication Série asynchrone (UART).
- Un Bus de communication synchrone I2C/SPI.

Entrées analogique

Alimentation

Entrées/sorties Numériques

UART

SPI/I2C

3. Alimentation

L'Arduino fournie à l'utilisateur une alimentation stabilisée pour le rôle d'alimenter les capteurs par exemple :

- 2 bornes d'alimentation : 5V et 3.2V.
- 3 bornes pour les masse GND.

4. Limitation de courant

La carte Arduino est sensible au courants entrants et sortant, il faut respecter les maximums pour la protéger :

- Le courant max recommandé sur une sortie Arduino est 20mA. En aucun cas il ne doit dépasser 40 mA.
- Le courant max que toutes les sorties réunies peuvent fournir ne doit pas dépasser 200 mA.
- Le courant max que toutes les sorties réunies peuvent recevoir ne doit pas dépasser 400mA.

5. Moyenne de transfert de programmes

La carte Arduino est composé d'un outil de transfert de programme :

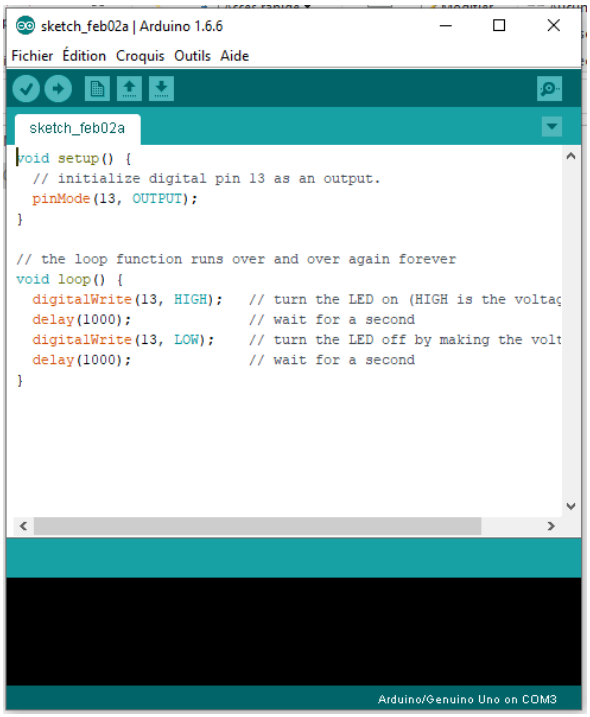
- Borne USB implanté dans la carte Arduino.
- Câble type B qui fait le transfert de programme depuis l'ordinateur vers la carte et ainsi que le transfert de données de la carte vers l'ordinateur.

6. Logiciel de programmation

Pour programmer l'Arduino, on dispose d'un environnement de développement intégré (IDE) constitué essentiellement de:

- Un éditeur de texte qui permet au programmeur de saisir son programme,
- Un compilateur C adapté au microcontrôleur ATMEGA.
- Outil de transfert de programme vers la carte.

7. Configurer L'IDE Arduino



La première fois qu'on Lance l'IDE, il faut faire une petite configuration qui consiste à:

- Choisir la carte Arduino sur laquelle on travaille: outils → type de carte → sélectionner votre carte dans la liste
- Définir le port COM sur lequel est connecté l'Arduino : outils → port → sélectionner le port COM. Si la prochaine fois, vous branchez votre Arduino sur un autre port USB, le plus souvent, le driver assigne un port COM différent, il faudra refaire cette étape.
- Choisir un dossier pour les programmes (croquis) que vous allez écrire:
- Utilisez l'explorateur windows pour créer un dossier de votre choix
- Dans l'IDE Arduino: fichier → préférences → Sélectionnez votre dossier dans le cadre Emplacement du carnet de croquis → OK.

III. Structure de programme Arduino

Il est très important de respecter certaines règles de programmation dans l'IDE. En général, elles se divisent en trois phases : la déclaration des variables locales, la configuration des périphériques de l'Arduino et la description du programme (algorithme).

Pour saisir ces procédures, nous nous appuierons sur l'exemple d'un algorithme de clignotement d'une LED branchée à la broche 13.

1. déclaration des variables locales

C'est le processus d'identification et de définition des variables utilisées dans un programme. Cela permet au compilateur de comprendre le type et la taille de la mémoire nécessaire pour stocker les valeurs de ces variables.

Dans ce programme la **broche 13** est nommée **led**, on utilise cette variable à la place de 13 afin **de simplifier la compréhension**.

2. Configuration des périphériques de l'Arduino

La configuration des périphériques est un processus de configuration des différents composants externes connectés à la carte Arduino. Cela inclut la définition des broches utilisées pour communiquer avec ces composants, la définition des paramètres de communication tels que la vitesse de transmission de données, ainsi que toute autre configuration nécessaire pour utiliser correctement ces périphériques.

Dans notre cas, la broche 13 doit être configurée en tant que sortie. La fonction qui permet cette configuration est **pinMode(broche, état).** La broche utilisée pour la LED est la broche 13, et elle est donc configurée en tant que sortie "OUTPUT".

3. Description du programme

La description du programme est le cœur du code et c'est là que le développeur exprime sa logique pour accomplir la tâche spécifiée. Le développeur décrit l'algorithme ou les étapes logiques que le programme doit suivre pour accomplir une tâche donnée.

```
void loop()
{
    digitalWrite(led, HIGH);      // allumer la LED ("HIGH" est la tension de sortie de 5V)
    delay(1000);                  // temporisation de 1000ms
    digitalWrite(led, LOW);       // atteindre la LED ("LOW" est la tension de sortie de 0V)
    delay(1000);                  // temporisation de 1000ms
}
```

Remarque : La phase 1 et 2 ne sont exécutées qu'une seule fois lors de l'alimentation de l'Arduino, tandis que la phase 3 continue d'être exécutée jusqu'à ce que l'alimentation soit coupée