Arduino – Controler une LED

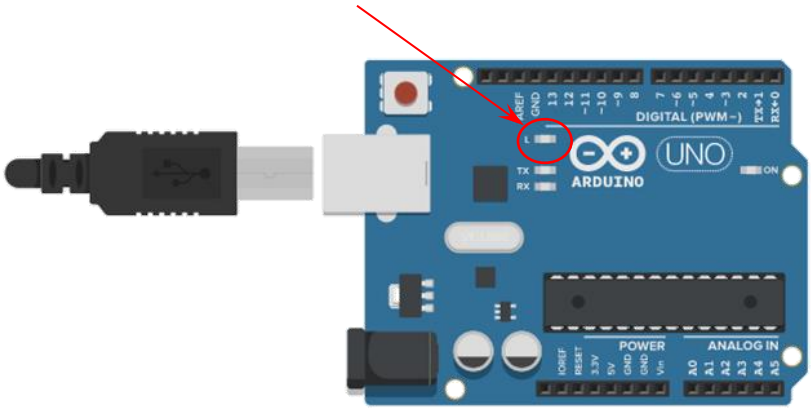
Introduction

L'Arduino est capable de piloter des Sorties. L'objet de cette session était d'allumer puis d'éteindre une LED par la programmation de l'Arduino.

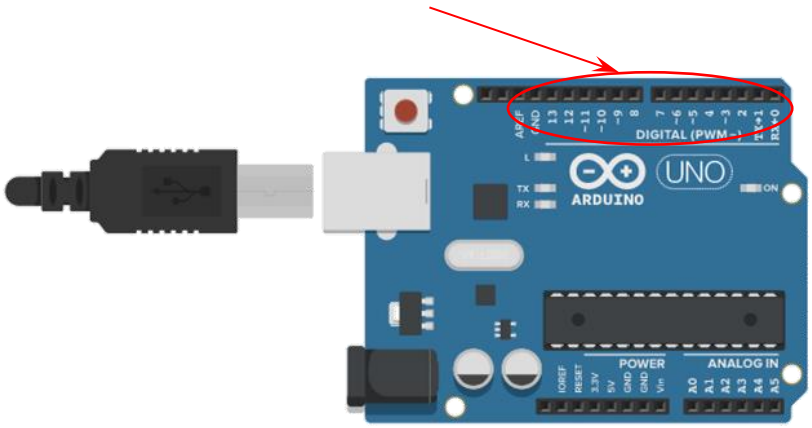
Etape 1 – Présentation des sorties de l'Arduino

Un voyant est intégré à l'Arduino, câblé sur la sortie 13, également nommée LED_BUILTIN.

Ce voyant se trouve ici :



De manière plus générale, les sorties disponibles sur l'arduino sont pilotable et se trouvent ici:



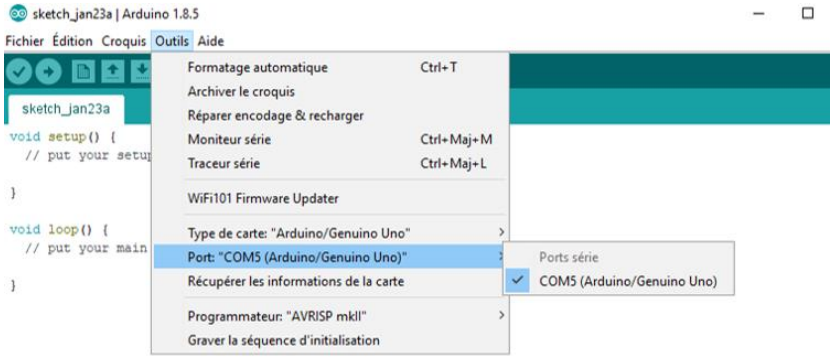
Etape 2 – Connecter l'Arduino

Pour piloter les sorties, il est nécessaire de créer un petit programme sur l'ordinateur. Puis de charger ce programme sur l'arduino.

La première étape est donc de connecter l'arduino à l'ordinateur via le câble USB.

Ensuite, lancer l'interface Arduino en cliquant sur l'icône :

L'interface se lance. Il faut ensuite s'assurer que l'Arduino est connecté. Aller dans le menu outils et vérifier que l'Arduino est bien reconnu (débrancher / rebrancher le câble USB si ce n'est pas le cas) :



Etape 3 – Piloter le voyant intégré

Un voyant est intégré à l'Arduino, câblé sur la sortie 13, également nommée LED_BUILTIN.

Afin de piloter ce voyant, il est nécessaire d'écrire quelques lignes de code simples :

Dans la rubrique void setup() { }, il faut déclarer le port que nous allons utiliser (LED_BUILTIN dans notre cas) avec la commande pinMode(sortie à piloter, Entrée ou sortie)

Info : entrée = INPUT et sortie = OUTPUT

Ici nous déclarons une sortie

Le code est donc le suivant :

```
void setup() {
  // put your setup code here, to run once:
  //Declarationde la sortie LED_BUILTIN comme sortie
  pinMode(LED_BUILTIN, OUTPUT);
}
```

Une fois la déclaration effectuée, il faut programmer les actions sur le voyant.

Pour ce faire nous utilisons la commande digitalWrite(Variable, Voltage)

Cette commande permet de piloter le courant alloué à la sortie Variable en modifiant la tension d'alimentation soit HIGH (allumé) soit LOW (éteint).

La variable que nous souhaitons piloter est la variable définie précédemment soit LED_BUILTIN.

La commande delay(temps) permet de faire un pause dans le programme, le temps est exprimé en millisecondes (1000 millisecondes = 1 seconde)

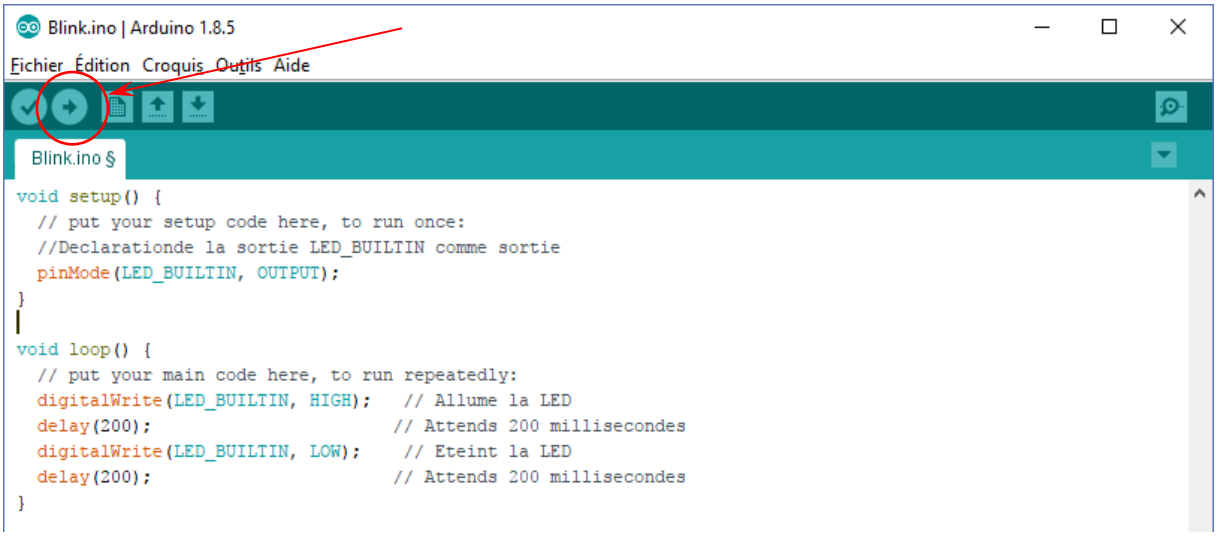
Ces commandes s'insèrent dans la section void loop() { } du programme.

Le programme ci-dessous permet d'allumer le voyant intégré à l'Arduino pendant 200 millisecondes, puis de l'éteindre pendant 200 millisecondes et de recommencer tant que l'arduino est connecté.

```
void setup() {
  // put your setup code here, to run once:
  //Declarationde la sortie LED_BUILTIN comme sortie
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED_BUILTIN, HIGH);   // Allume la LED
  delay(200);                       // Attends 200 millisecondes
  digitalWrite(LED_BUILTIN, LOW);    // Eteint la LED
  delay(200);                       // Attends 200 millisecondes
}
```

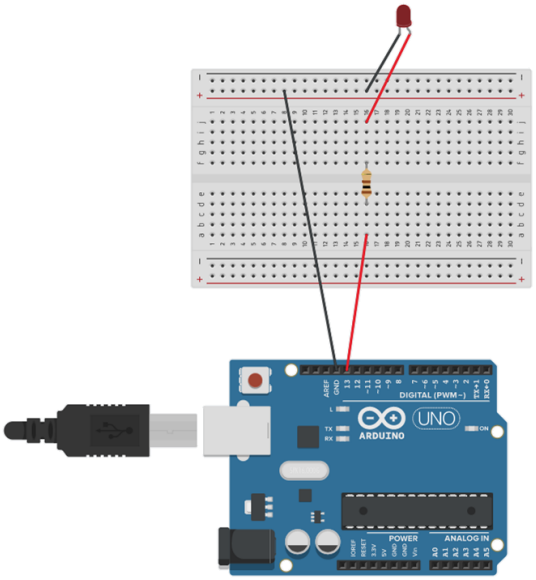
Une fois le programme écrit, il n'y a plus qu'à téléverser le programme sur l'Arduino et admirer le voyant qui clignote. Cette opération s'effectue par le bouton représentant une flèche :



Etape 4 – Piloter une LED

Pour aller plus loin, il est possible de piloter une (ou plusieurs) LED.

Tout d'abord il faut effectuer le montage suivant :



La broche 13 de l'arduino est connectée sur la breadboard. Ensuite une résistance de 100 ohms (Ω) est mise en place pour éviter que la LED ne soit traversée par un courant trop important. Puis la LED est connectée. Enfin le retour à la masse se fait en connectant un dernier fil sur la broche identifiée GND de l'arduino.

La broche 13 correspondant à la même sortie que le voyant intégré, le programme précedent fait clignoter de façon identique le voyant et la LED.

Etape 5 – Aller plus loin

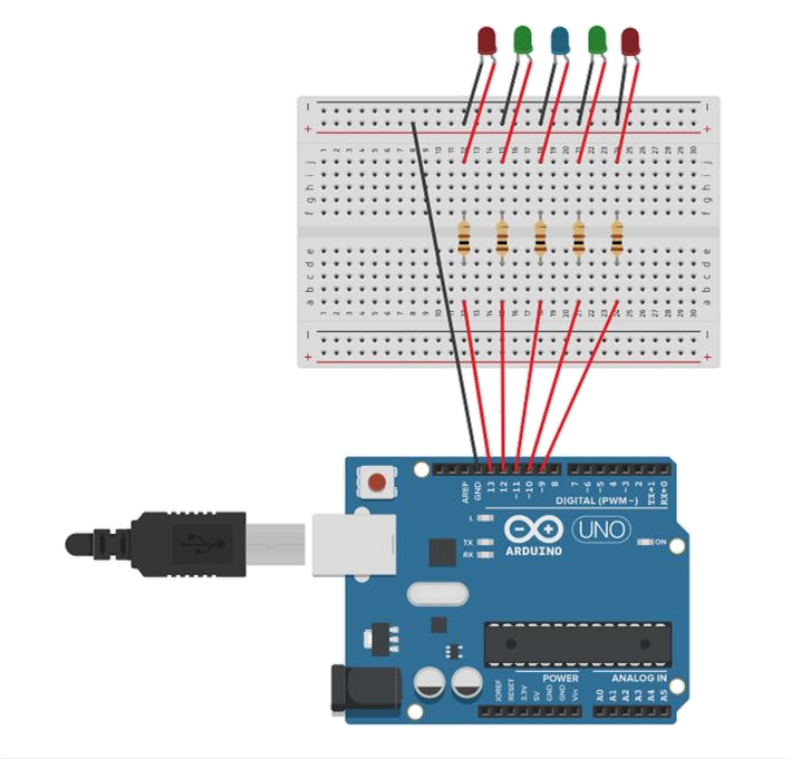
Pour aller plus loin, libre à vous de rajouter des LED (respecter la mise en place de résistance en face de chaque LED). La déclaration des sorties se fait simplement en multipliant les déclarations pinMode(sortie à piloter, Entrée ou sortie). La sortie pilotée correspond au numéro de la broche de l'Arduino sur laquelle vous avez branché votre circuit (entre 1 et 13).

Dans le programme principal il faut modifier / ajouter des commandes digitalWrite(Variable, Voltage) où la variable est le numéro de broche que vous venez de déclarer avec la commande pinMode.

Exemple :

Le montage et le programme suivant permettent de faire clignoter les LED par un balayage continu dans un sens puis dans l'autre :

Montage



Programme


Blink.ino | Arduino 1.8.5
Fichier Édition Croquis Outils Aide
Blink.ino
void setup() {
// put your setup code here, to run once:
//Declarationde la sortie LED_BUILTIN comme sortie
pinMode(LED_BUILTIN, OUTPUT);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
digitalWrite(LED_BUILTIN, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(LED_BUILTIN, LOW); // Eteint la LED
digitalWrite(12, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(12, LOW); // Eteint la LED
digitalWrite(11, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(11, LOW); // Eteint la LED
digitalWrite(10, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(10, LOW); // Eteint la LED
digitalWrite(9, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(9, LOW); // Eteint la LED
digitalWrite(10, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(10, LOW); // Eteint la LED
digitalWrite(11, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(11, LOW); // Eteint la LED
digitalWrite(12, HIGH); // Allume la LED
delay(200); // Attends 200 millisecondes
digitalWrite(12, LOW); // Eteint la LED
}
Téléversement terminé
Le croquis utilise 1114 octets (3%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2039 octets pour les variables
37
Arduino/Genuino Uno sur COM5