



Module SCI

Ecole Nationale Supérieure d'Informatique d'Alger (Ex. ini)

Compte rendu TP1

3 Mars 2021

Introduction

La plateforme électronique Arduino est un environnement très intéressant de développement IoT, fournissant un ensemble de composants (mini-contrôleur, pins de branchement, LEDs, ...etc) pour implémenter des solutions IoT.

Une première étape d'apprentissage serait de se familiariser avec cet outil, son utilisation et comment le programmer. Ce présent rapport sert à résumer mon expérience durant le premier LAB du module.

Objectifs

1. **Comprendre la plateforme Arduino** : présenter les différents composants et fonctionnement de l'IDE
2. **Tester un premier programme** : Test avec la LED interne au board avec un programme exemple : blink
3. **Breadboard et jeu de lumière** : Découvrir l'extension avec le lab d'essai

La plateforme Arduino

Arduino est une plateforme dédiée à la création de solutions digitales à base de composants électroniques à moindre coût et destiné à un large public même ceux issus des domaines par forcément d'ingénierie électronique.

Plus précisément, Arduino est un micro-contrôleur : un circuit intégré compact fabriqué pour exécuter une opération dans un système embarqué, il inclut généralement un processeur, une mémoire, les entrées/sorties ... etc

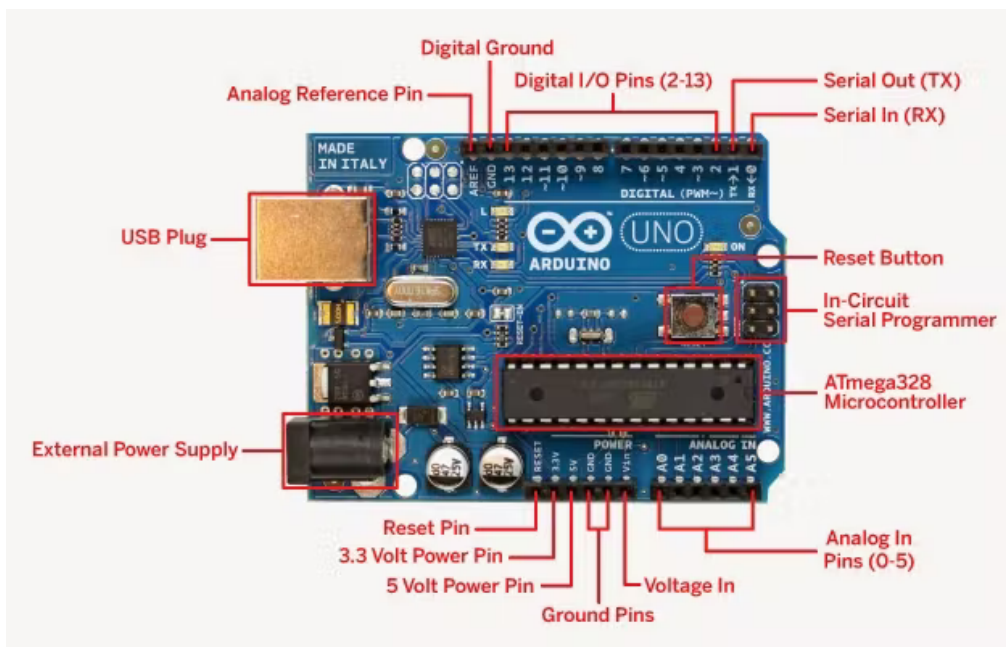


Figure 01 : Architecture d'une plateforme Arduino



Figure 02 : Arduino MKR WI-FI 1010

Le cœur d'un Arduino est le MCU (Microcontroller Unit) qui s'est chargé de l'exécution. La plateforme peut être alimentée à l'aide d'une source d'énergie externe (7 - 12V DC) ou un câble USB (5V)

Étapes du lab :

1. Installation de l'IDE Arduino

Nous avons d'abord installé l'IDE pour écrire nos codes, ensuite nous avons choisi le package requis pour notre modèle Arduino MKR 1010 WI-FI.

Download the Arduino IDE



Figure 3 : Installation de l'IDE

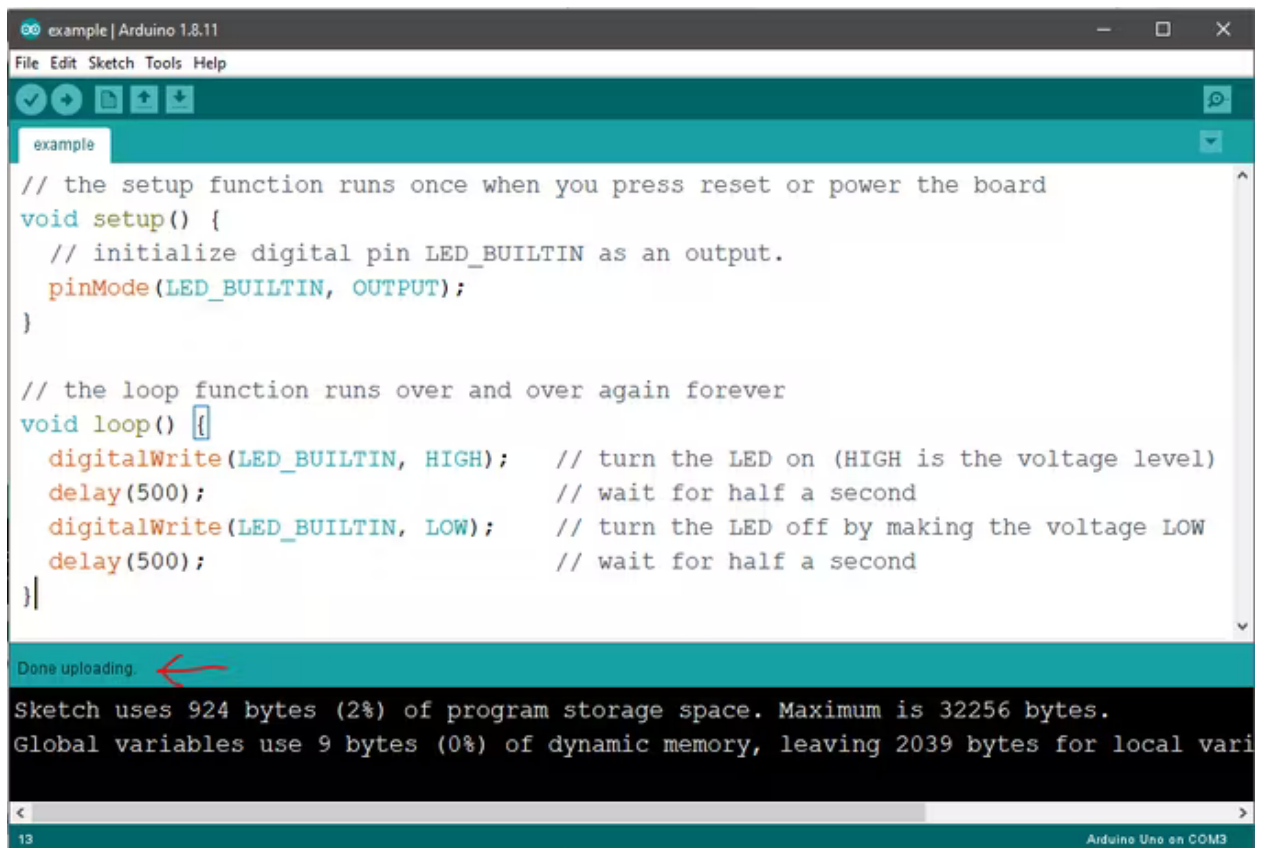
2. Branchement à l'ordinateur via câble USB-C

Pour savoir si la plateforme est bien connectée, on peut se référer aux lampes LED internes. Ensuite on sélectionne le port COM5 au niveau de l'IDE (Tools> Port)

3. Programme exemple (blink)

Nous sélectionnons un exemple prêt "Blink" (File>Examples>Basics>Blink), on le compile d'abord ensuite on le verse pour qu'il soit exécuté

Remarque : Nous apercevons le clignotement de la LED interne, on peut jouer avec les valeurs du delay pour changer la vitesse de cette dernière.



The screenshot shows the Arduino IDE interface. The title bar reads "example | Arduino 1.8.11". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area contains the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(500);                      // wait for half a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(500);                      // wait for half a second
}
```

Below the code editor, a status bar indicates "Done uploading." with a red arrow pointing to it. Below that, a black box displays memory usage information: "Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables." The bottom status bar shows "13" on the left and "Arduino Uno en COM3" on the right.

Figure 4 : Code du programme de test "Blink"

Analysons le code ci-dessus, On remarque qu'on définit la fonction `setup()` qui est censée être exécutée, la fonction `pinMode(pin, mode)` permet de définir quelle pin sera utilisée et est-ce qu'elle est en mode entrée ou sortie.

La fonction `digitalWrite(pin, voltage_level)` permet d'allumer la LED lorsque le `voltage_level = 'High'`, l'éteindre sinon.

4. Utilisation du breadboard (Lab d'essai)

Ce module permet d'éteindre les pins de notre Arduino, il contient deux rangées horizontalement liées indiquées en générale par des couleurs (rouge/bleu)(+/-) destinée à l'alimentation. et une large rangée de pins connectées verticalement mais séparées par une zone, pour éviter le court-circuit.

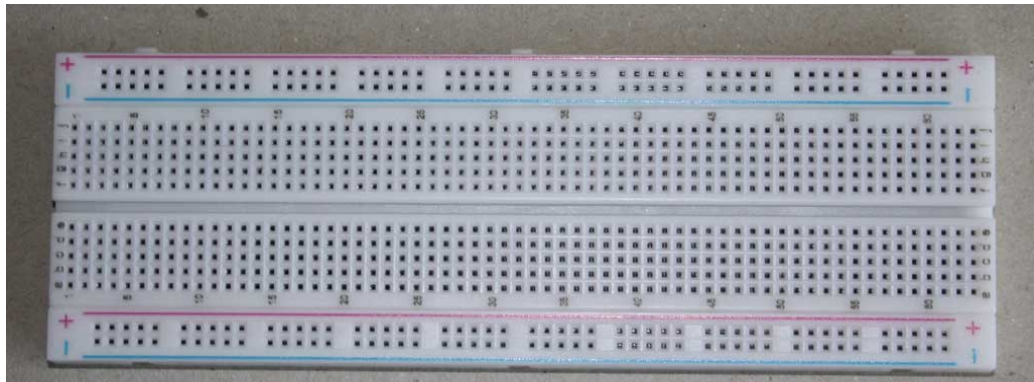


Figure 5 : Le lab d'essai (Breadboard)

Nous allons l'exploiter pour faire fonctionner une LED externe. Voici le diagramme du circuit :

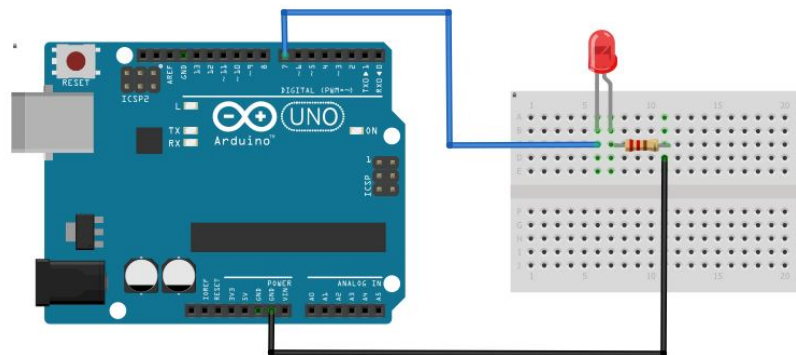


Figure 6 :Circuit d'allumage pour une seule LED

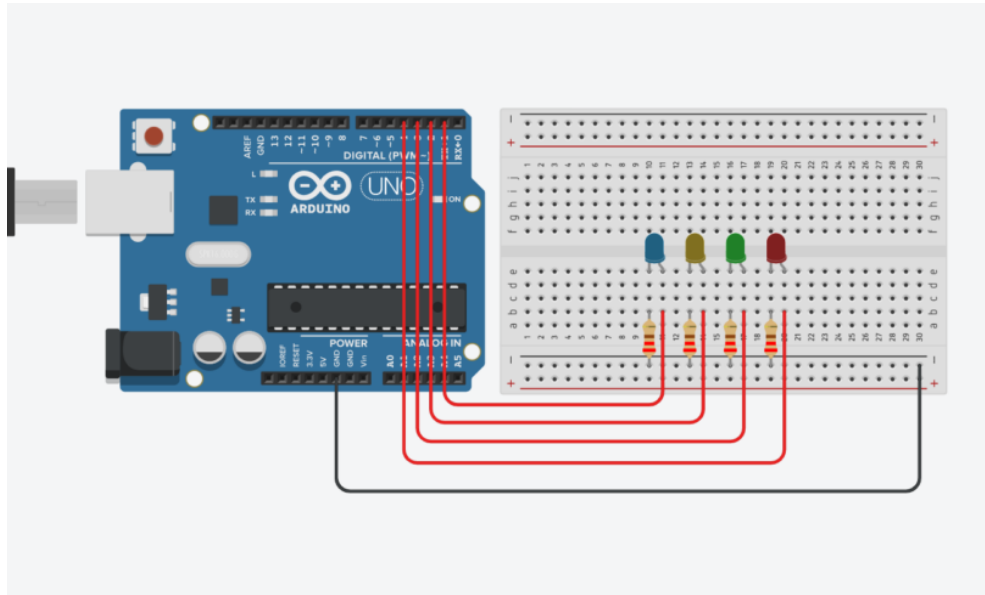


Figure 7 :Circuit d'allumage pour 3 LEDs

Nous avons besoin de :

- Lampe externe LED
- 2 câbles
- Une résistance 330 ohm

On change la pin dans le code, au lieu d'utiliser la PIN_BUILTIN (qui est par défaut la 6) on utilise la 7 .

Résultat obtenu :

On a bien eu un allumage de la LED externe, pour pousser un peu plus notre compréhension surtout du câblage du circuit, nous avons utilisé 3 LEDs externes pour réaliser un jeu de lumière en alternant l'allumage de ces dernières.

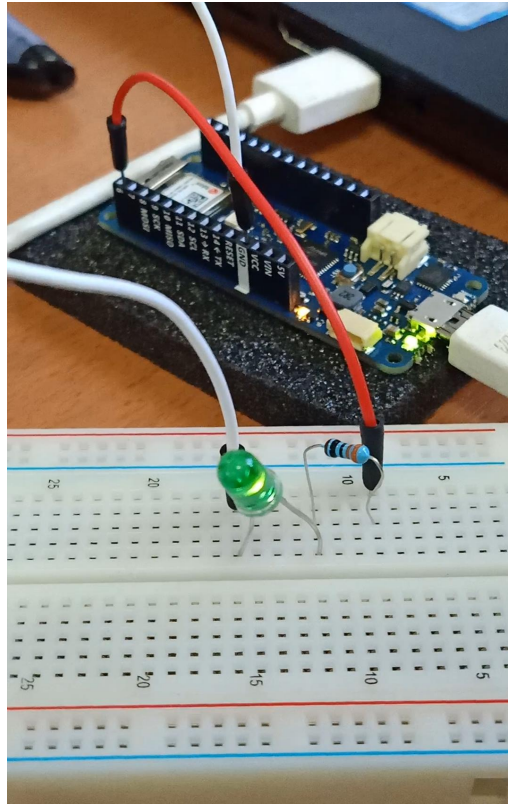
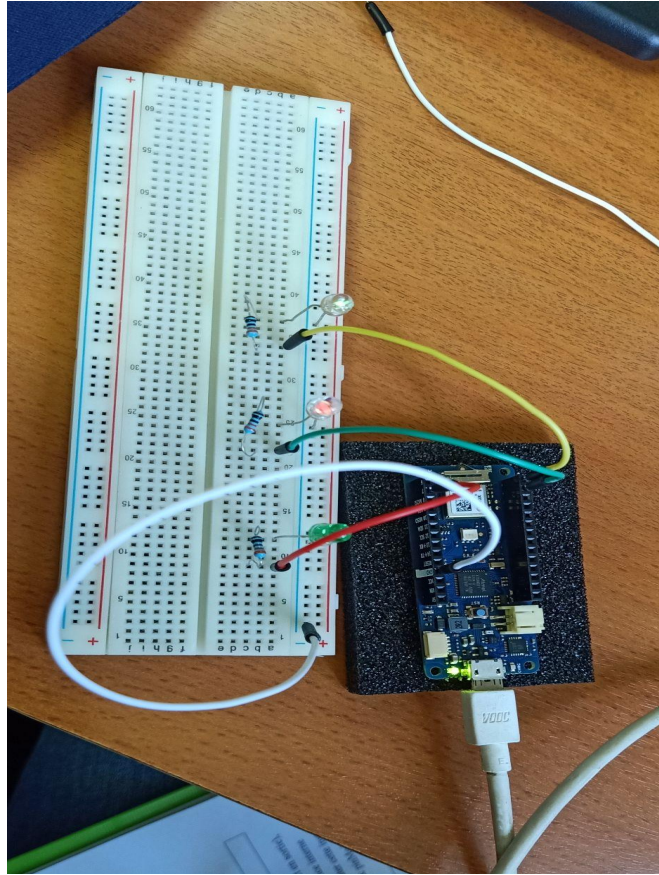


Figure 8 : Résultat avec une seule LED



Figuer 9 : Résultat avec 3 LEDs

Conclusion :

Durant cette première partie nous avons exploité notre plateforme à l'aide d'un câble, la suite portera sur la manipulation à l'aide du Wi-Fi .