



**Faculté des Sciences et Technologie**  
**Département d'électrotechnique et d'automatique**  
**Niveau : L3 Automatique**  
**Module : Automates Programmables Industriels**  
**Année Universitaire : 2021/2022**

## **TP 1 : Simulation et réalisation d'un système automatisé simple**

**L'objectif :** est de prendre en main le matériel et logiciel pour réaliser un système automatisé simple:

### **I. Matériel Nécessaire :**

- Un ordinateur
- Une carte Arduino UNO
- Un câble USB Ordinateur <-> Carte Arduino
- Des LEDs
- Des résistances de  $220\Omega$

### **II. Description du fonctionnement du système :**

Ce premier programme a pour but de vérifier le bon fonctionnement de l'ensemble, notamment la communication entre l'IDE (Integrated Development Environment, ou environnement de développement : c'est le logiciel utilisé). Pour l'instant, c'est le câble USB qui alimente la carte. Le programme est écrit dans l'IDE, puis téléversé dans la carte Arduino. Le résultat sera de faire clignoter une LED de couleur quelconque sur la plaque d'essai. Il vise également le bon fonctionnement du programme ISIS proteus en ajoutant les bibliothèques (bibliothèques) nécessaires au fonctionnement de ce système.

Il faut, en premier lieu, programmer le microcontrôleur, pour qu'il fasse ce qu'on veut

Le CPU du microcontrôleur interprète le programme présent dans la mémoire flash.

Le programme est en langage machine (du binaire). Le langage machine dépend du microcontrôleur.

#### **1. Travail demandé :**

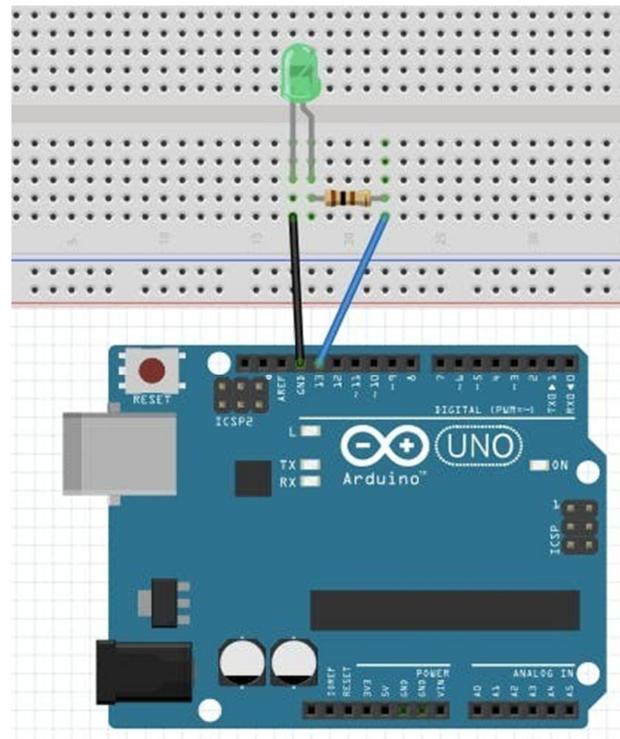
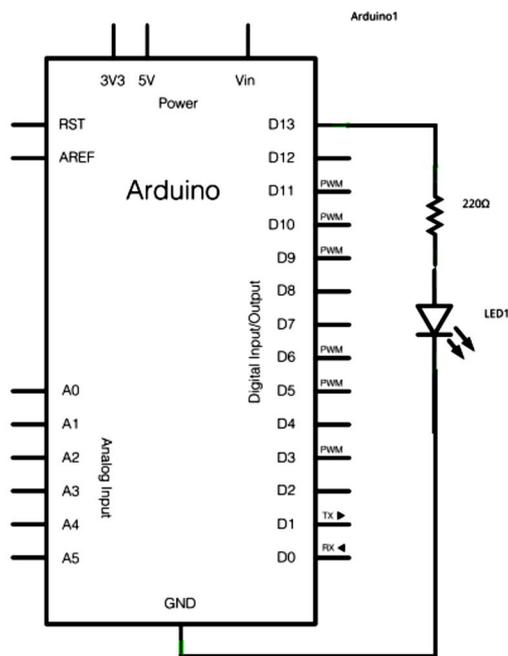
##### **a) Avant de commencer:**

- Certains composants électroniques sont fragiles
- Notamment ils n'aiment pas : l'électricité statique, être branchés à l'envers ou recevoir plus de tension (volts).
- Évitez de toucher les pattes des circuits intégrés
- Revérifiez que tout est bien branché avant d'alimenter le circuit

## b) Pratique:

### 1. faire clignoter une LED

- (Installez et) Lancez l'interface Arduino
- Créez un nouveau programme (sketch)
- Entrez le code donné
- Simulez le schéma sous ISIS proteus
- Branchez la LED sur l'Arduino (en série avec la résistance) comme indiqué
- Branchez l'Arduino au PC par l'USB
- Téléversez, et la LED devrait clignoter
- Modifiez le programme pour faire d'autres effets



Tous les programmes sont composés au minimum de deux fonctions :

**void setup()** et **void loop()**

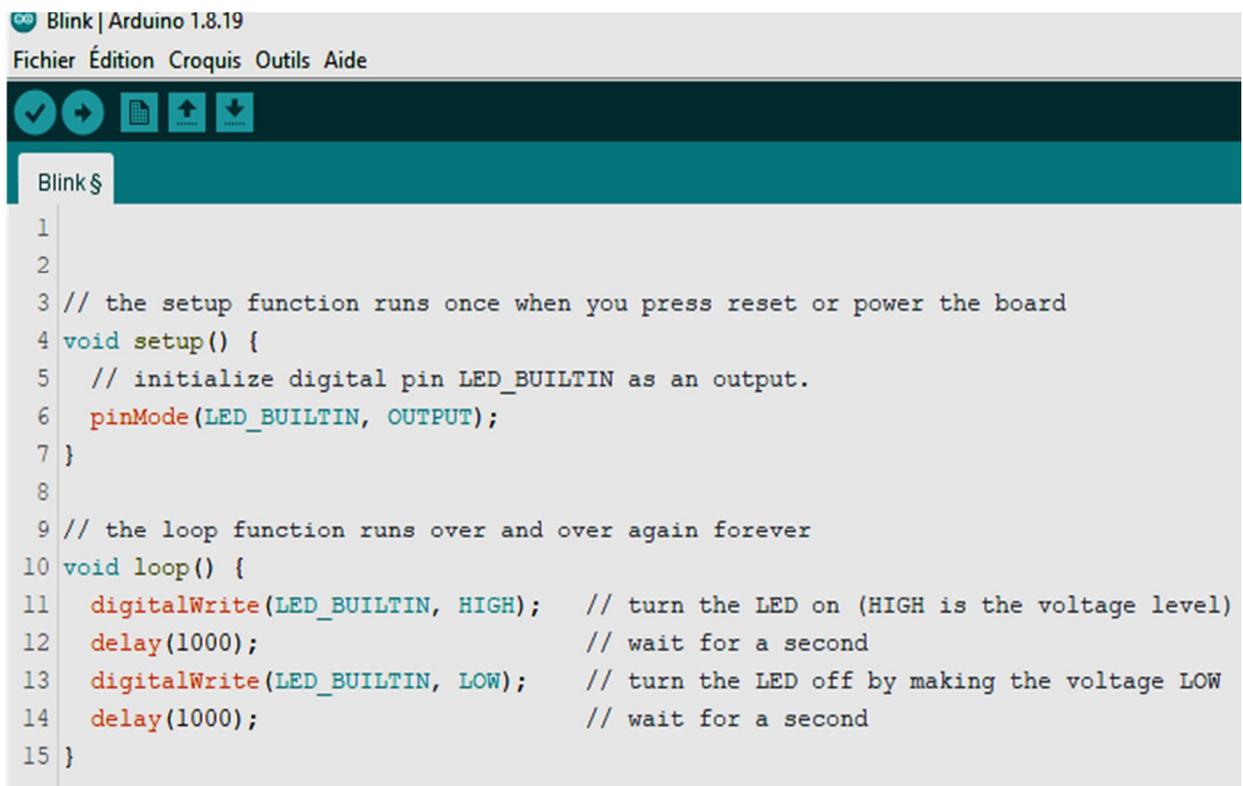
La première exécute une seule fois les instructions situées entre les accolades. Elle sert le plus souvent pour initialiser des variables. La deuxième répète indéfiniment (loop = boucle) les instructions situées entre les accolades. Dans ce programme, l'instruction située dans le setup :

**pinMode(13, OUTPUT);** indique que la broche repérée 13 est considérée comme une sortie. (Le programme en lui-même est situé dans la fonction « loop » : il consiste à mettre une

tension haute sur la broche 13 pendant un délai de 1000 millisecondes, puis une tension basse (égale à zéro) pendant un délai de 1000 millisecondes:

```
digitalWrite(13,HIGH);  
delay(1000);  
digitalWrite(13,LOW);  
delay(1000);
```

On peut, par exemple, faire clignoter la DEL deux fois plus vite en modifiant les deux valeurs delay, pour prendre la valeur 500 au lieu de 1000



```
Blink | Arduino 1.8.19  
Fichier Édition Croquis Outils Aide  
Blink $  
1  
2  
3 // the setup function runs once when you press reset or power the board  
4 void setup() {  
5   // initialize digital pin LED_BUILTIN as an output.  
6   pinMode(LED_BUILTIN, OUTPUT);  
7 }  
8  
9 // the loop function runs over and over again forever  
10 void loop() {  
11   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
12   delay(1000); // wait for a second  
13   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
14   delay(1000); // wait for a second  
15 }
```

## 2. faire varier graduellement la luminosité d'une LED

Un signal PWM est une sortie du microcontrôleur qui permet, par exemple, de commander un hacheur, varier la vitesse d'un moteur à courant continu. En général, la fréquence est fixée et on fait varier le rapport cyclique du signal.

- **Arduino et PWM**

L'IDE Arduino possède une fonction intégrée "analogWrite()" qui peut être utilisée pour générer un signal PWM. La fréquence de ce signal généré pour la plupart des broches sera d'environ 490Hz et nous pouvons donner la valeur de 0-255 en utilisant cette fonction.

**analogWrite(0)** signifie un signal d'un rapport cyclique de 0%.

**analogWrite(127)** signifie un signal avec un rapport cyclique de 50%.

**analogWrite(255)** signifie un signal avec un rapport cyclique de 100%.

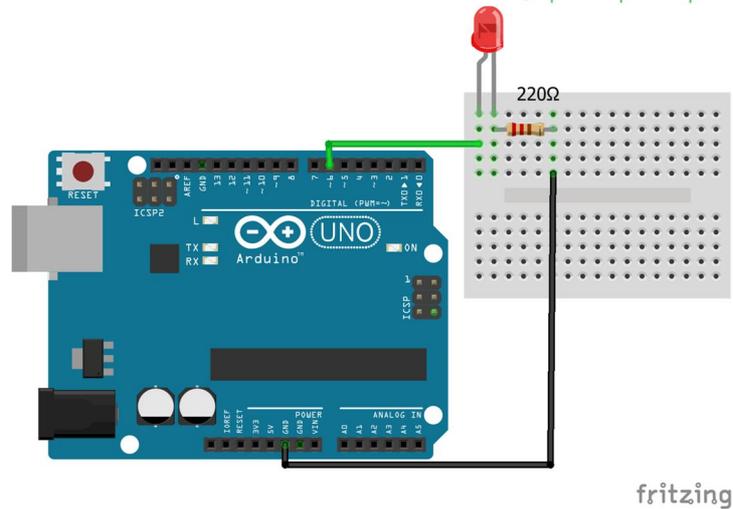
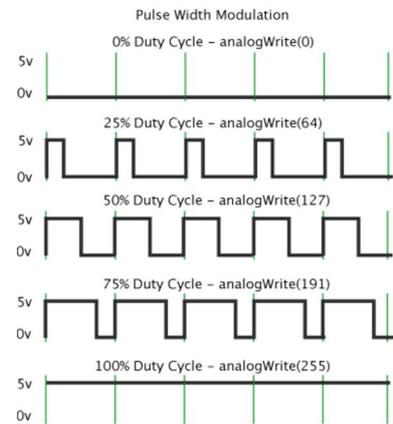
Sur Arduino Uno, les broches PWM sont 3, 5, 6, 9, 10 et 11. La fréquence du signal PWM sur les broches 5 et 6 sera d'environ 980Hz et sur les autres broches sera de 490Hz. Les broches PWM sont marquées du signe ~.

**Syntax :**

`analogWrite(pin, value)`

Effectuez les connexions comme décrit ci-dessous.

Connectez la branche positive de la LED à la broche numérique 6 de l'Arduino. Ensuite, connectez la résistance de 220 ohms à la branche négative de la LED et connectez l'autre extrémité de la résistance à la broche de masse de l'Arduino.



Vous pouvez utiliser le programme fading dans le menu Arduino IDE:

Fichier>Exemples->Analog.