

ELE3312
Microcontrôleurs et applications
Laboratoire 1

Auteur : Jean Pierre David

Introduction

Les microcontrôleurs sont omniprésents dans la société industrialisée. Leur puissance de calcul ne cesse d'augmenter alors que leurs prix sont incroyablement bas. Si on dispose déjà d'un ordinateur, il est aujourd'hui possible de se procurer un environnement complet de développement pour quelques dizaines de dollars. L'environnement *Arduino* est un environnement de développement particulièrement prisé car il est accessible à toute personne disposant de quelques connaissances de programmation. Il est à la base de nombreux projets amateurs et même industriels. Ce premier laboratoire veut vous faire goûter le plaisir et la facilité de développer des petites applications pour microcontrôleur même si on n'a aucune idée de la manière dont ils fonctionnent. Bien sûr, nous n'en resterons pas là. Au cours des laboratoires suivants, vous comprendrez de mieux en mieux ce qui se passe réellement au cœur d'un microcontrôleur et vous apprendrez comment en tirer un maximum de performances.

Objectifs

1. Installer et configurer votre environnement de développement *Arduino* sur votre ordinateur personnel.
2. Comprendre les manipulations de base avec le kit de laboratoire.
3. Concevoir des applications relativement simples pour vous familiariser avec :
 - a. Les sorties binaires
 - b. Les sorties PWM
 - c. Les entrées analogiques
 - d. La communication série pour le débogage

Mise en garde importante

qui pourrait vous épargner des heures de trouble 😊

Ces laboratoires sont initialement écrits pour la carte NUCLEO-F446RE,
qui est la carte recommandée pour ces laboratoires.

Les laboratoires peuvent aussi fonctionner avec d'autres cartes NUCLEO
moyennant quelques modifications.

!!! Assurez-vous de faire les changements, si nécessaire !!!

1^{ère} partie : préparation à la maison

Mode opératoire

Installation de ST-LINK

Pour programmer votre carte de développement de type STM32 NUCLEO (typiquement la carte NUCLEO-F446RE), vous aurez besoin de l'utilitaire de programmation développé par le manufacturier.

1. Allez à l'adresse suivante : http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stsw-link009.html
2. Téléchargez l'utilitaire STSW-LINK009 au bas de la page (bouton « Get software »). Si ce n'est déjà fait, vous devrez vous enregistrer sur le site du manufacturier.

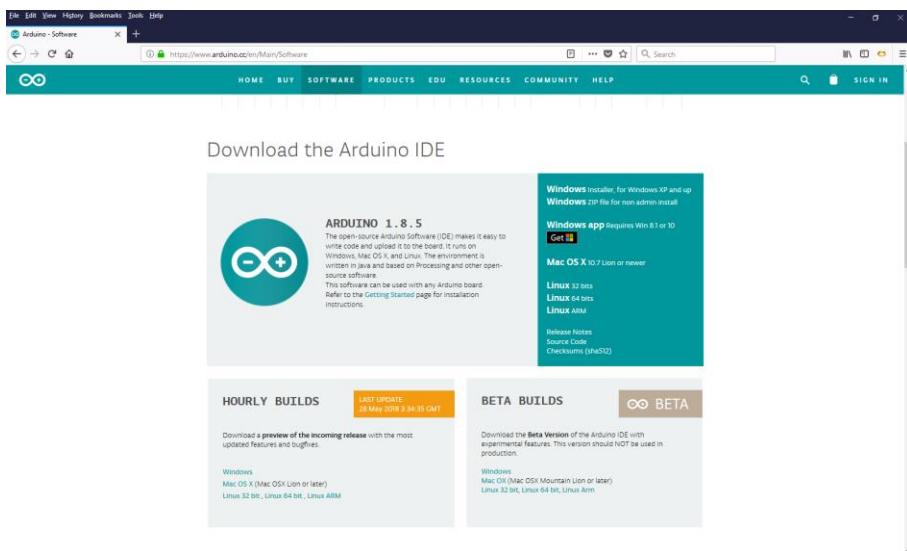
The screenshot shows a web browser window with the title "STSW-LINK009 - ST-LINK, ST-LINK". The address bar shows the URL "www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-utilities/stsw-link009.html". The page content is organized into several sections:

- Product Specifications:** A table with columns for Description, Version, and Size. It lists "DB2694: USB driver for ST-LINK/V2 and ST-LINK/V2-1" (Version 3.0, 111 KB).
- Legal:** A table for the License Agreement with columns for Description, Version, and Size. It lists "SLA0047: Image V2 - SOFTWARE LICENSE AGREEMENT" (Version 1.16, 99 KB).
- GET SOFTWARE:** A table with columns for Part Number, Software Version, Marketing Status, Supplier, and Order from ST. It shows "STSW-LINK009" with "2.0.0" as the Software Version, "Active" as the Marketing Status, and "ST" as the Supplier. A blue "Get Software" button is located at the bottom right of this table.

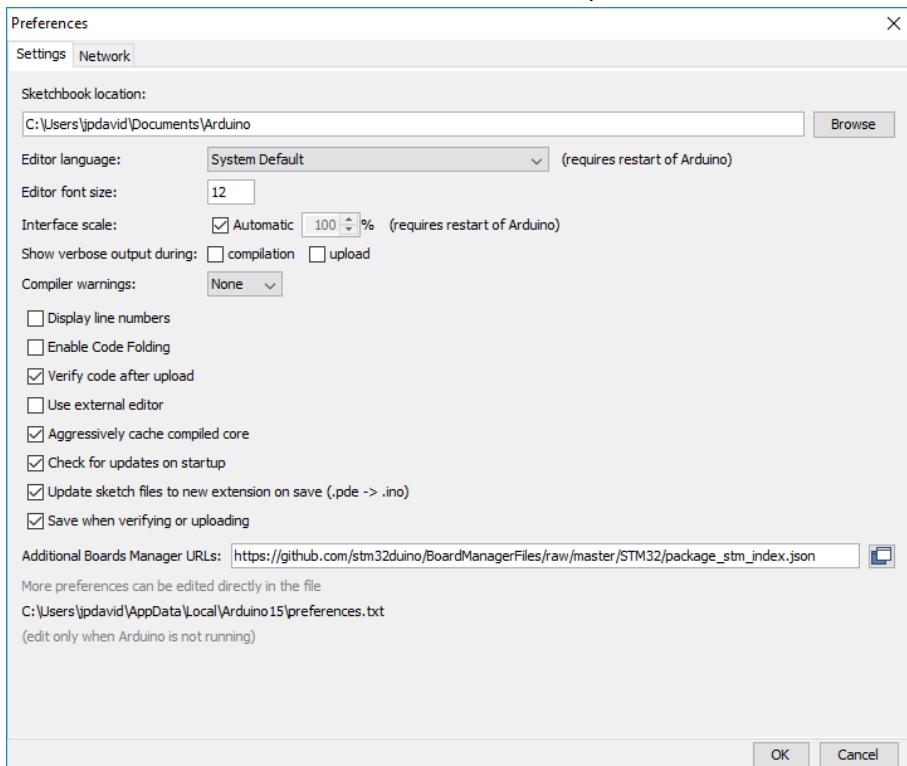
3. Suivez les instructions pour installer le logiciel sur votre ordinateur et acceptez l'installation de tous les pilotes qu'il vous propose.

Installation de l'environnement Arduino

1. Allez à l'adresse suivante : <https://www.arduino.cc/en/Main/Software>
2. Téléchargez « Arduino IDE », typiquement en utilisant le lien vers « Windows installer »



3. Suivez les instructions pour installer le logiciel sur votre ordinateur et acceptez l'installation de tous les pilotes qu'il vous propose.
4. Démarrez l'IDE *Arduino* et allez dans le menu File/Preferences :

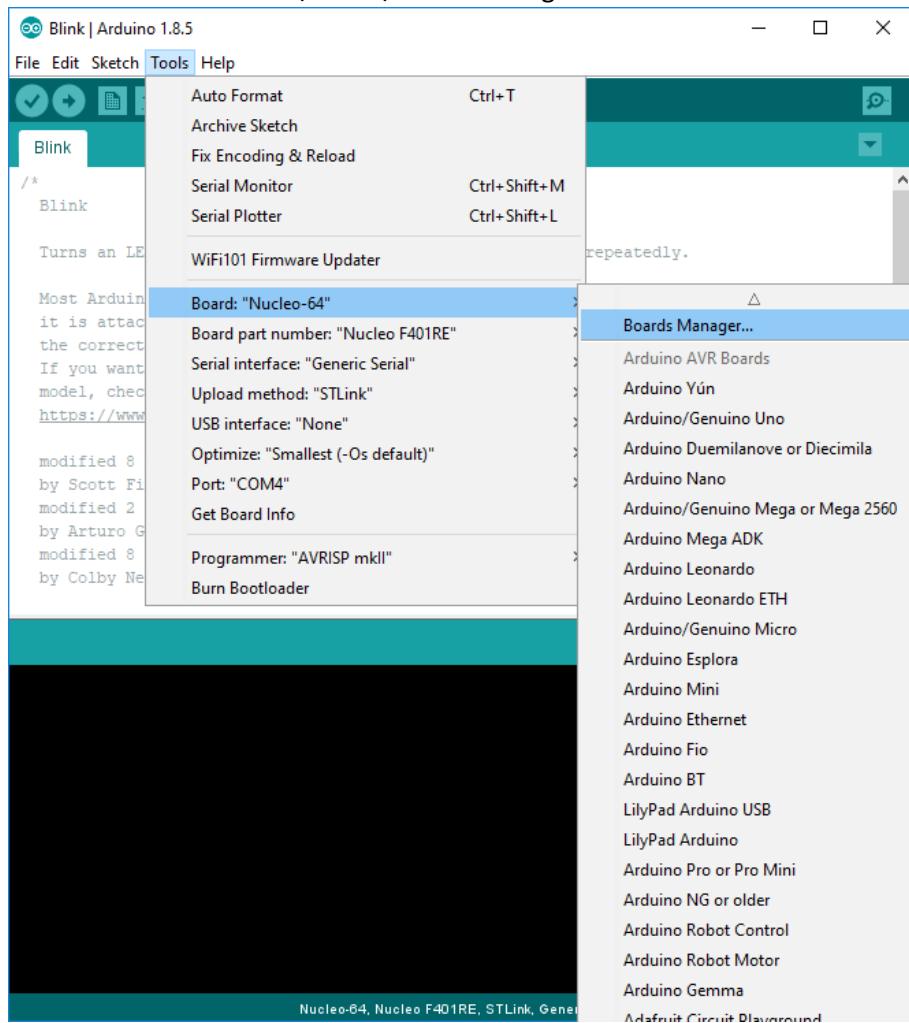


5. Copiez-collez l'adresse suivante dans le cadre intitulé « Additional Boards Manager URLs » :

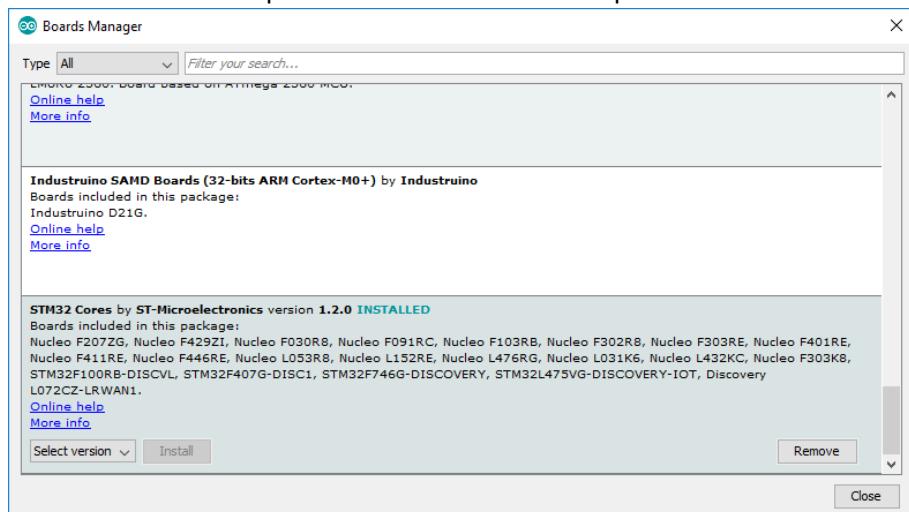
https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json

6. Validez avec le bouton « OK »

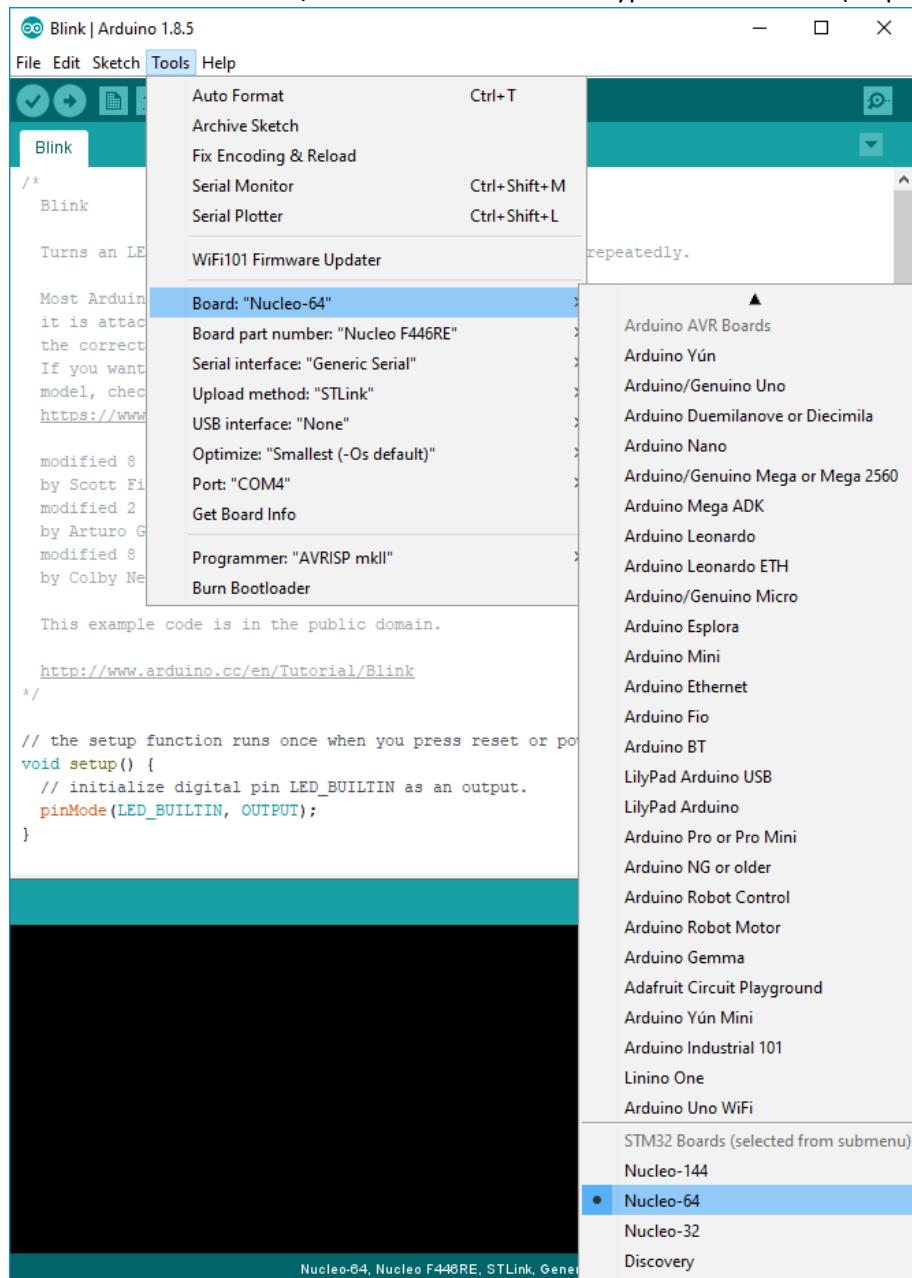
7. Allez dans le menu Tools/Board/Board Manager...



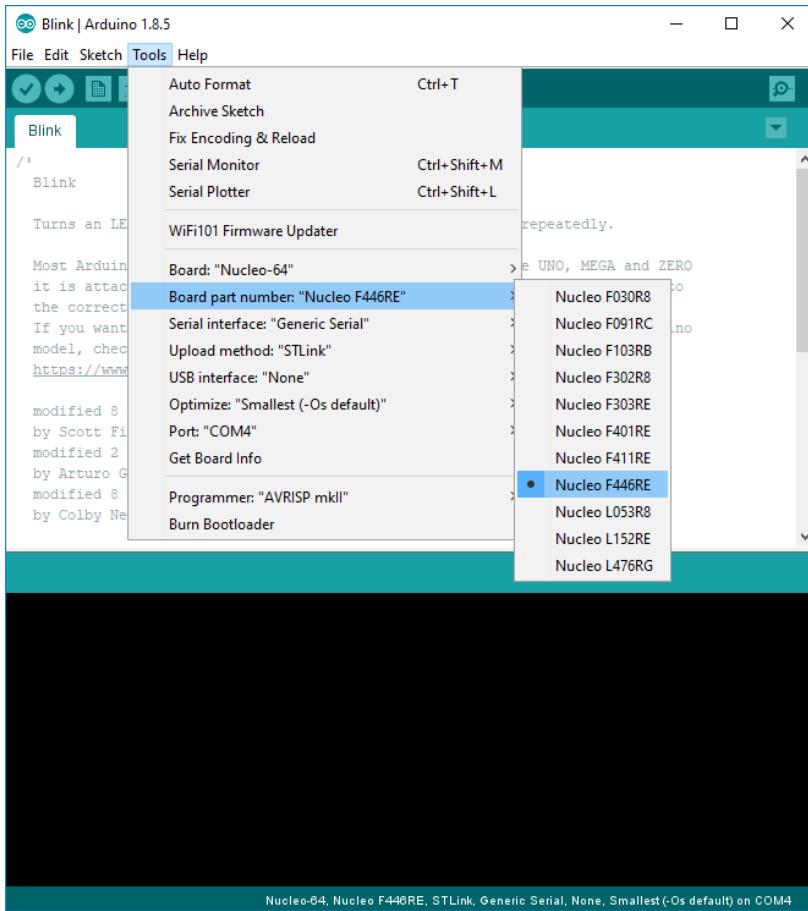
8. Tout en bas, cliquez sur STM32 cores ... et lancez l'installation des cartes STM32 en cliquant sur le bouton « Install » après avoir choisi la version la plus récente.



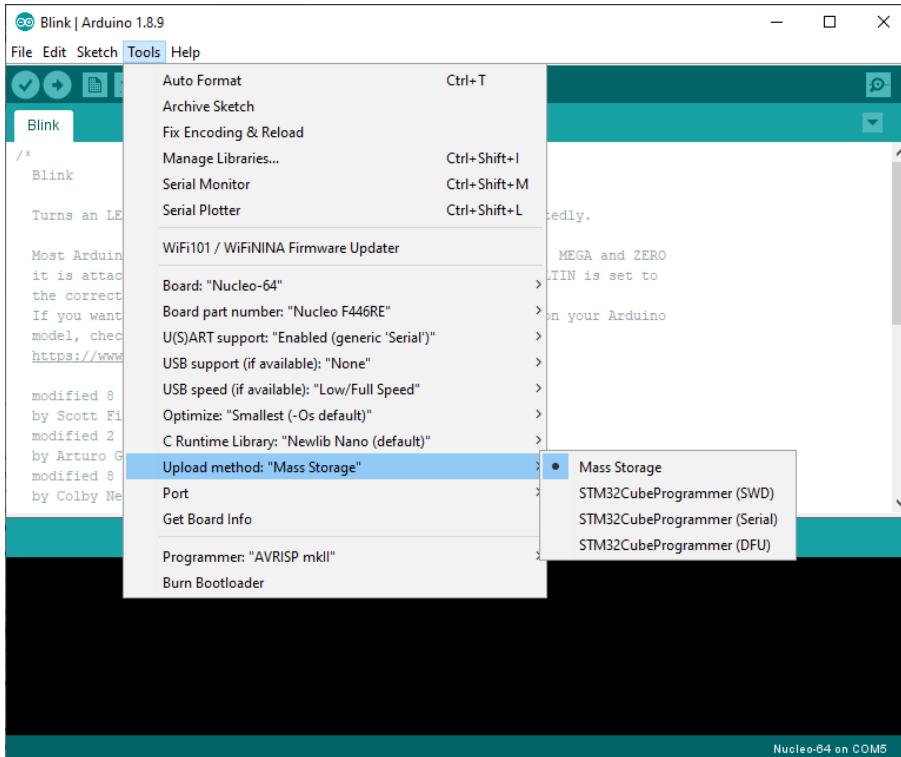
9. Allez dans le menu Tools/Board et sélectionnez le type de votre carte (en principe NUCLEO-64)



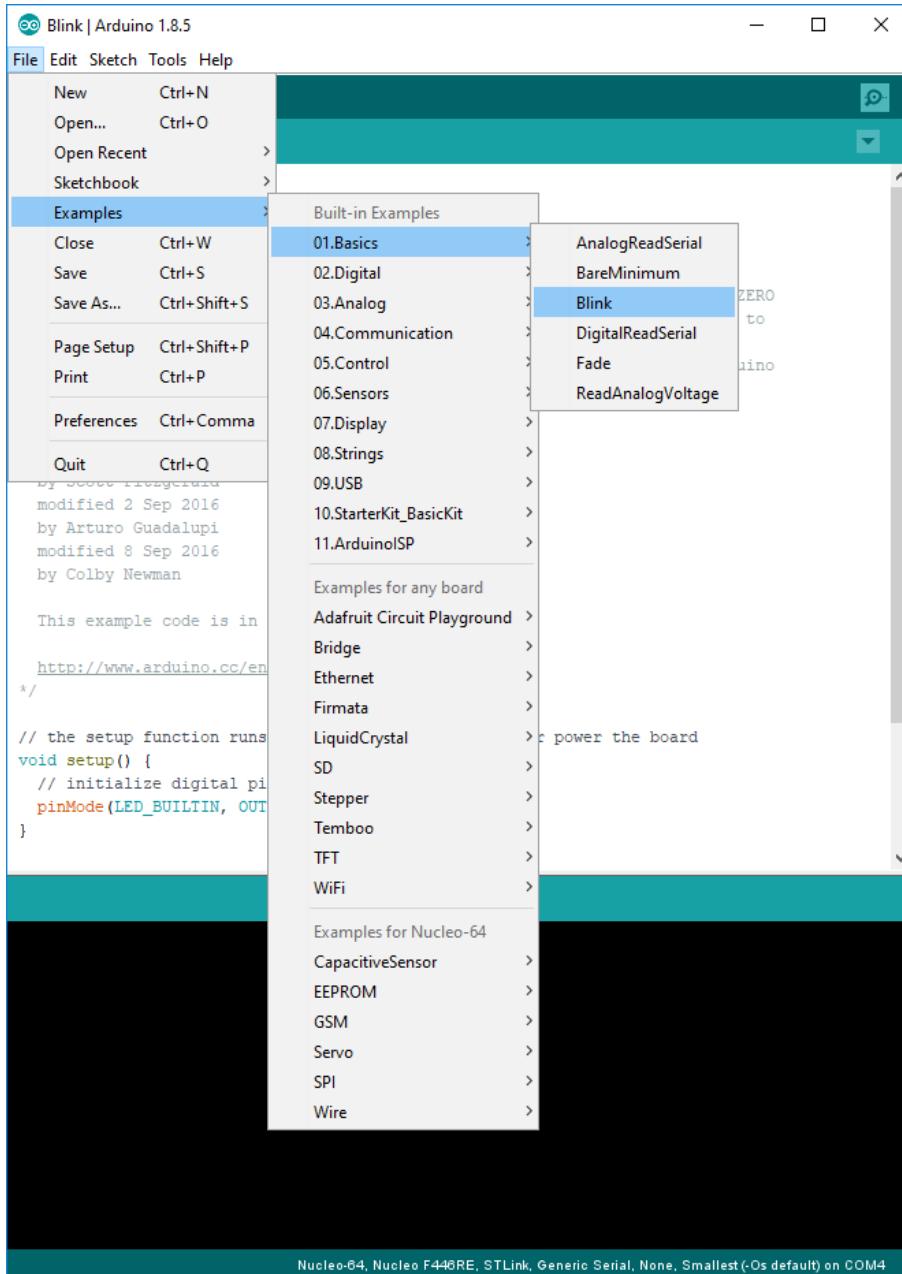
10. Allez dans le menu Tools/Board part number et choisissez votre carte (en principe NUCLEO-F446RE)



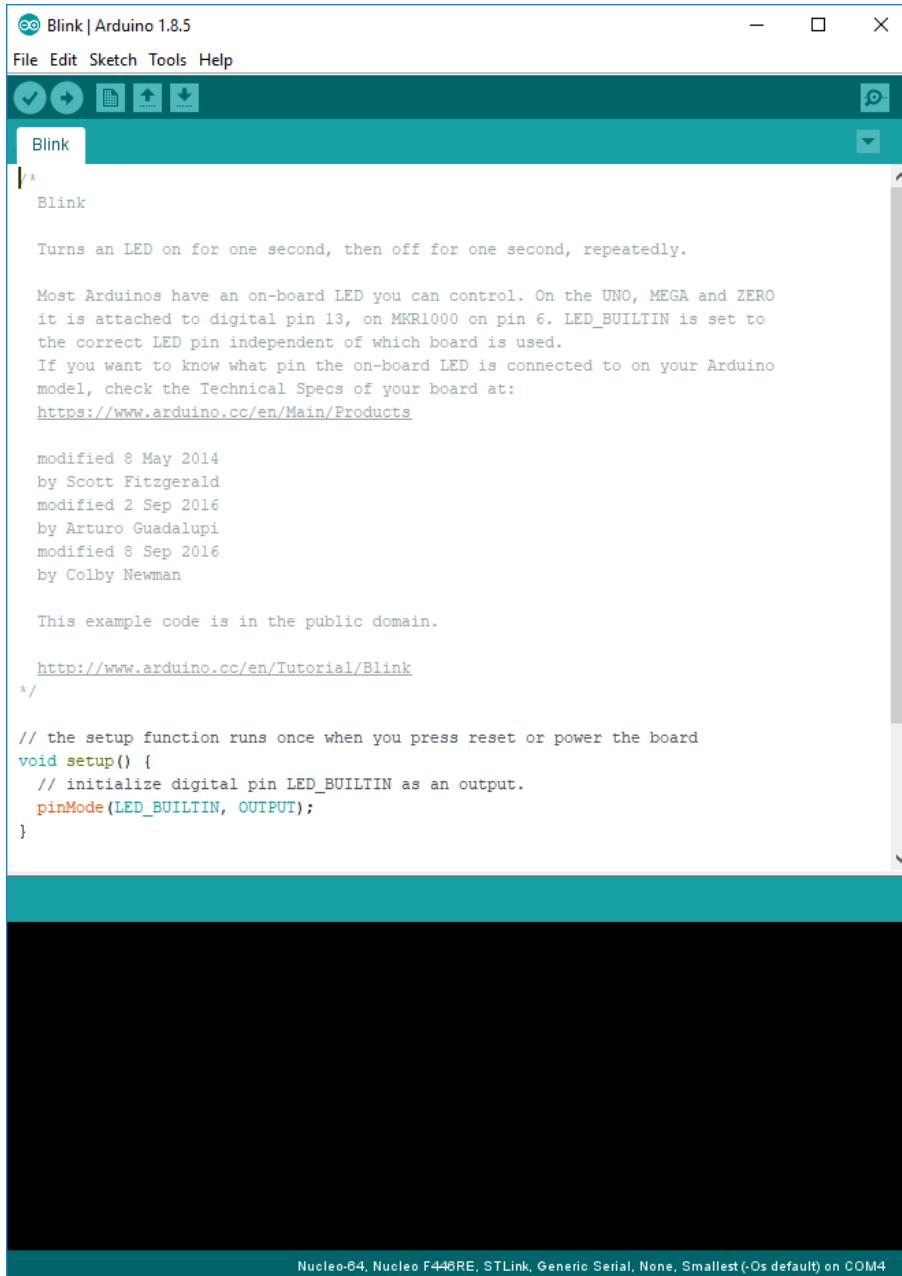
11. Allez dans le menu Tools/Upload method et choisissez le mode Mass Storage



12. Allez dans le menu File/Examples/01.Basics/Blink pour sélectionner une application de référence



13. Une nouvelle fenêtre *Arduino* s'est ouverte avec l'application « Blink ». Vous pouvez fermer la précédente.



14. Sauvegardez cette application dans le répertoire de votre choix (Menu File/Save as)
15. Fermez l'application, redémarrez votre ordinateur et ouvrez l'IDE *Arduino* à nouveau. Vous devriez retrouver votre environnement tel que vous l'avez configuré avec l'application *Blink* pré-chargée. Sinon, ouvrez l'application *Blink* que vous avez sauvegardée
16. Connectez votre carte à l'ordinateur au moyen du câble USB
17. Compilez votre programme au moyen du bouton en dessous du menu File



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Shows "Blink | Arduino 1.8.5".
- Menu Bar:** Includes "File", "Edit", "Sketch", "Tools", and "Help".
- Toolbar:** Features icons for file operations (New, Open, Save, Print, Upload, Download) and a pin configuration icon.
- Sketch Editor:** Displays the "Blink" sketch code:

```
/*
Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
```
- Compile Status:** Shows "Done compiling."
- Serial Monitor:** Displays the output of the compilation:

```
Archiving built core (caching) in: C:\Users\jp david\AppData\Local\Temp\arduino_cache_346418\
Sketch uses 12700 bytes (2%) of program storage space. Maximum is 524288 bytes.
Global variables use 1052 bytes (0%) of dynamic memory, leaving 130020 bytes for local varia
```
- Bottom Status:** Shows "Nucleo-64, Nucleo F446RE, STLink, Generic Serial, None, Smallest (-Os default) on COM4".

18. Téléversez votre programme dans la carte au moyen de la flèche 

19. Contemplez votre œuvre, la diode verte clignote ! Si ce n'est pas le cas, repassez à travers toutes les étapes pour trouver ce qui a pu poser problème.

Remarque

L'IDE Arduino est articulé autour de deux méthodes : *setup()* et *loop()*.

La méthode `setup()` est appelée une seule fois au tout début. On l'utilise généralement pour configurer les entrées/sorties et initialiser les variables.

La méthode *loop()* est une boucle infinie. Cela veut dire que dès qu'elle se termine, elle est rappelée automatiquement indéfiniment.

Découverte du kit de laboratoire

La carte à microcontrôleur

Votre carte NUCLEO-F446RE est alimentée par le port USB (5V) toutefois, un convertisseur soudé sur la carte produit le 3.3V nécessaire au microcontrôleur. Vous trouverez toute la documentation technique de la carte au lien suivant : <http://www.st.com/en/evaluation-tools/nucleo-f446re.html>

! ATTENTION !

Les broches du microcontrôleur ne supportent que des signaux entre 0V et 3.3V.

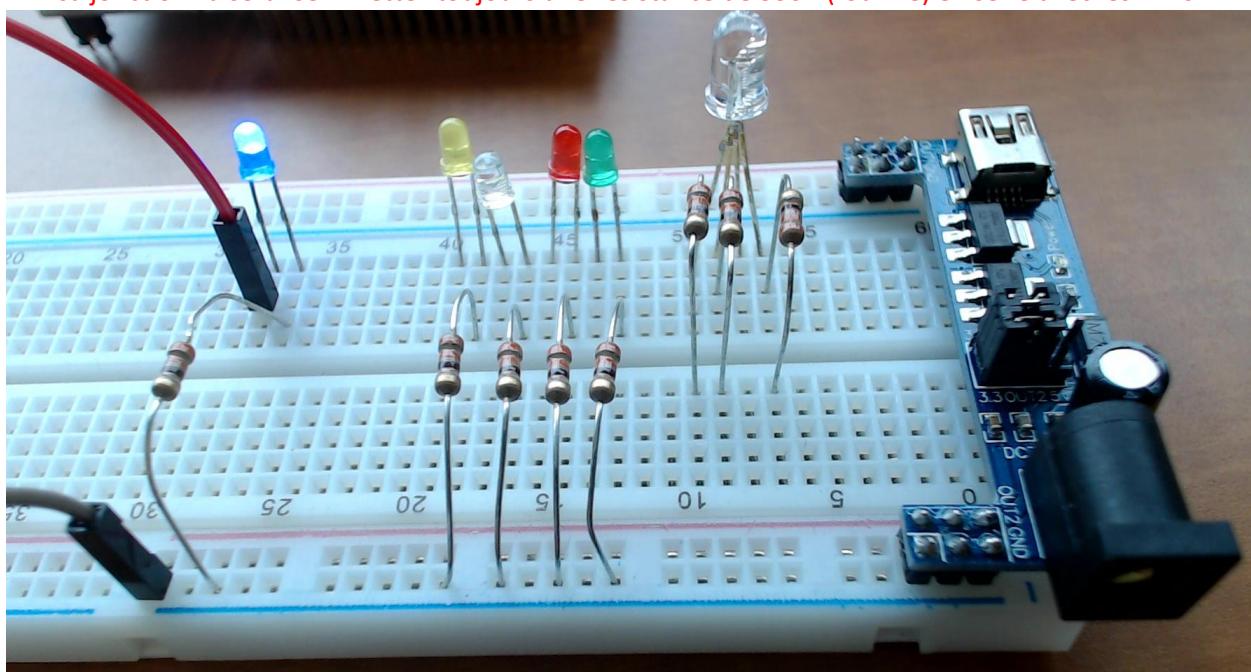
Il ne faut jamais connecter des signaux 5V à la carte

Les diodes électroluminescentes (DEL)

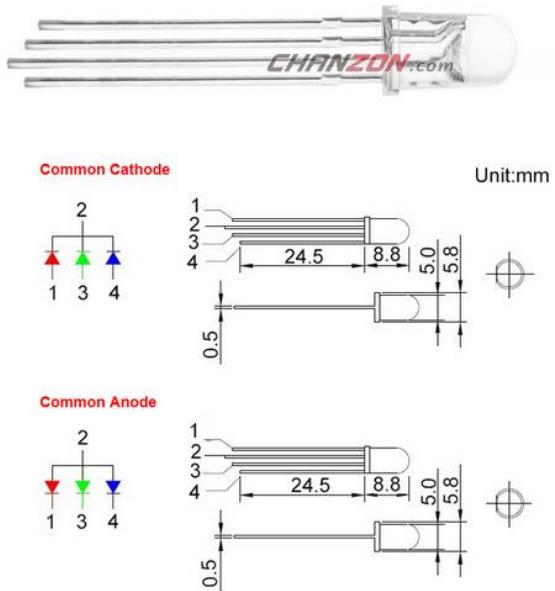
Votre kit comprend des DELs monochromes et des DELs RGB.

! ATTENTION !

Une diode doit être alimentée en courant. Si vous envoyez directement une tension 3.3V sur une diode, sa jonction va se briser. Mettez toujours une résistance de 330R (fournie) en série avec les DELs.

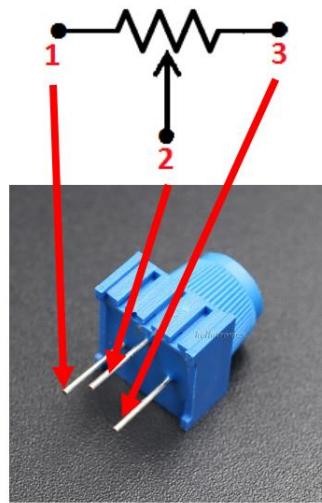


Remarque : Les DELs sont polarisées. Le courant doit entrer par l'anode et ressortir par la cathode. Pour les DELs monochromes, **l'anode est associée à la broche la plus longue**. En principe, les DELs RGB que vous avez ont leur anode commune (sur la broche la plus longue également), comme indiqué sur la figure ci-dessous. Il faut remarquer que dans ce cas, on doit connecter l'anode au 3.3V et chacune des cathodes à une résistance 330R. La diode s'allumera lorsqu'on présentera une tension nulle à l'autre borne de chacune des résistances. Si vous n'êtes pas certain de votre modèle, essayez d'allumer la composante rouge (pin 1-2) dans un sens et puis dans l'autre, toujours avec une résistance de 330R en série. Lorsque la diode s'allume, le courant entre par l'anode (plus positive) et ressort par la cathode.



Les potentiomètres

Vous disposez de deux potentiomètres de 10K chacun. Ces potentiomètres sont utiles pour générer une tension variable entre 0V et 3.3V. Connectez les broches externes (#1 et #3 ci-dessous), une à la masse et l'autre au 3.3V. Vous pouvez alors faire varier la tension de la broche centrale (#2 ci-dessous) en faisant tourner le bouton du potentiomètre :

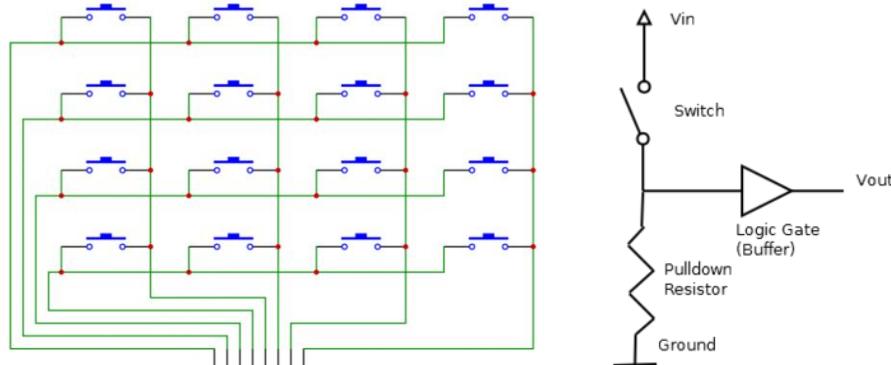


Remarquez que cela ne fonctionne correctement que si aucun courant ne sort de la broche centrale. C'est pratiquement le cas lorsqu'on la connecte à une broche d'entrée du microcontrôleur car elles ont une impédance très élevée.

Le clavier

Un petit clavier de 16 touche vous est fourni. Lorsqu'on pèse sur une touche, un court-circuit se produit entre la ligne et la colonne correspondantes (voir figure ci-dessous). En observant le circuit imprimé du clavier, vous découvrirez quelle broche est connectée à quelle ligne et quelle colonne. Pour produire un

signal de 3.3V lorsqu'on pèse sur un interrupteur, on le connecte habituellement en série avec une résistance pull-down :



Nous verrons plus tard dans le cours comment utiliser un clavier au complet.

Connecteurs audio

Ce composant sera détaillé plus tard.

Module ultrasons

Ce composant sera détaillé plus tard.

Écran TFT

Ce composant sera détaillé plus tard.

Premiers pas avec l'environnement Arduino

Expérience 1

En sachant que la DEL verte de la carte est aussi connectée à la sortie D13, faites clignoter la DEL bleue en même temps que la DEL verte de la carte.

Pour cela, identifiez d'abord la sortie D13 de la carte. Utilisez un fil pour connecter cette sortie à l'anode de la DEL bleue sur le breadboard. Connectez finalement une résistance de 330R entre la masse et la cathode de la DEL bleue. Pour rappel, votre code devrait être en ce moment :

```
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);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
}
```

Expérience 2

Faites clignoter la DEL bleue en opposition de phase avec la DEL verte de la carte.

La DEL doit donc s'allumer lorsque la sortie D13 est à 0V. Pour cela, connectez la sortie D13 à la cathode de la diode bleue sur le breadboard et connectez son anode à **une résistance de 330R dont l'autre extrémité est connectée au 3.3V**. Pour obtenir le 3.3V, vous pouvez directement utiliser la broche du connecteur qui porte ce nom (3V3) sur la carte du microcontrôleur. De cette manière, lorsque D13 est à 0V, un courant circulera dans la diode bleue sur le breadboard tandis que lorsque D13 est à 3.3V, un courant circulera dans la diode verte de la carte.

Expérience 3

Faites clignoter la DEL sur la sortie D7.

Pour cela, vous devez recompiler le code dans l'IDE *Arduino* en mentionnant que vous voulez la pin 7 à la place de la constante *LED_BUILTIN*. Vous devez donc faire trois remplacements :

```
pinMode (7, OUTPUT);  
digitalWrite (7, HIGH);  
digitalWrite (7, LOW);
```

Pour recompiler et téléverser, appuyez sur la petite flèche  comme précédemment.

Expérience 4

Faites varier la puissance de la DEL au moyen d'un PWM.

Essayez le code suivant :

```
void setup () {  
    pinMode(6, OUTPUT);  
}  
  
void loop () {  
    analogWrite(6, 255);  delay(1000);  
    analogWrite(6, 127);  delay(2000);  
    analogWrite(6, 0);   delay(3000);  
}
```

Assurez-vous de connecter l'anode de votre DEL au signal D6 et sa cathode vers la masse à travers une résistance de 330R.

Vous devriez voir la DEL à pleine puissance pendant 1000ms, à demi puissance pendant 2000ms et finalement éteinte pendant 3000ms. Il faut comprendre que le microcontrôleur envoie un signal carré de type PWM. Donc, en fait, la diode clignote très rapidement mais votre œil n'est pas capable de le percevoir et il pense que c'est son intensité qui varie, dépendamment du rapport cyclique du PWM. Pour en savoir plus sur le rapport cyclique, vous pouvez aller voir le lien ci-dessous :

https://fr.wikipedia.org/wiki/Rapport_cyclique

Expérience 5

Faites varier la puissance de la DEL au moyen d'un potentiomètre.

Pour cela, le microcontrôleur va lire la tension du potentiomètre et l'utiliser pour contrôler la puissance du PWM. Branchez le potentiomètre entre la masse et le 3.3V pour avoir une tension réglable sur la broche du milieu et connectez cette tension à l'entrée analogique A0 du microcontrôleur. Ensuite, essayez le code suivant :

```
int x = 0;

void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(6, OUTPUT);
}

void loop() {
    x = analogRead(0);
    analogWrite(6, x/4);
}
```

Il faut noter que la fonction *analogRead* retourne une valeur entre 0 (0V) et 1023 (3.3V). Donc, on est obligé de diviser cette valeur par 4 quand on veut spécifier la valeur du PWM qui doit être comprise entre 0 et 255. Vérifiez que lorsque vous tournez le potentiomètre, vous ajustez l'intensité de la DEL en conséquence.

Expérience 6

Parcourir toutes les couleurs RGB possibles.

Pour cela, connectez l'anode de la diode RGB au +3.3V. Ensuite, **connectez chacune des cathodes en série avec une résistance de 330R**, Finalement, connectez l'autre borne de chaque résistance aux signaux D3, D5 et D6. Essayez le code suivant :

```
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(3, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
}

void loop() {
    for (int i=0; i <= 255; i+=16) {
        analogWrite(3, i);
        for (int j=0; j <= 255; j+=16) {
            analogWrite(5, j);
            for (int k=0; k <= 255; k+=16) {
                analogWrite(6, k);
                delay(250);
            }
        }
    }
}
```

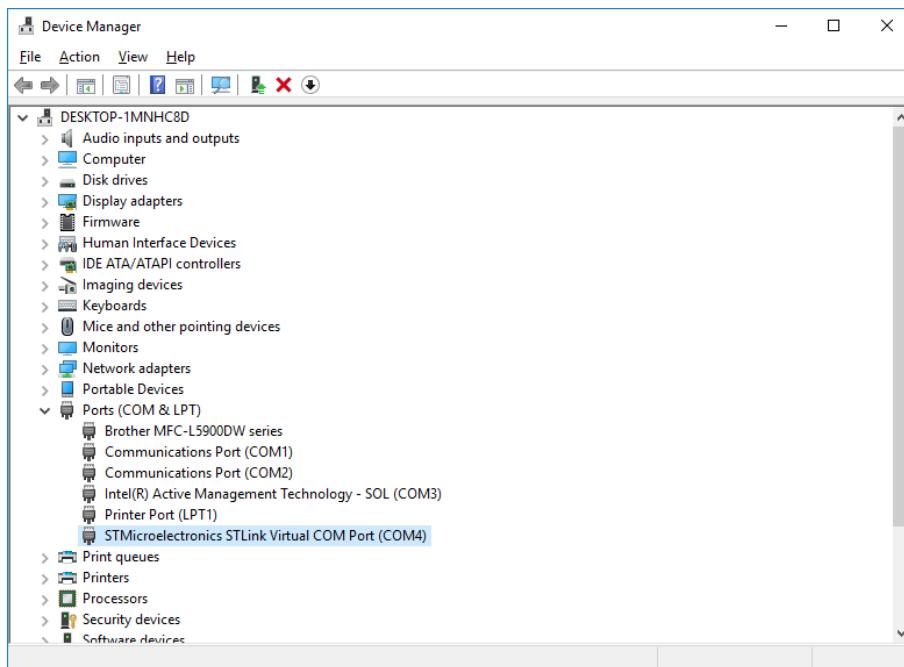
```
    }  
}  
}
```

Vous devriez voir la DEL changer de couleur toutes les 250ms.

Expérience 7

Envoyer des messages à la console du PC

Pour cela, vous devez configurer le port de communication dans le menu Tools/Port de l'IDE Arduino. Sélectionnez le port qui correspond au STLINK Virtual COM (souvent c'est le numéro le plus élevé). Pour trouver le port, vous pouvez aussi lancer l'application « Device Manager » de Windows (voir ci-dessous) :



Ensuite, dans l'IDE, ouvrez l'outil Tools/Serial Monitor. Tous les messages provenant de la carte apparaîtront dans cette fenêtre. Essayez maintenant d'ajouter des messages dans le programme précédent :

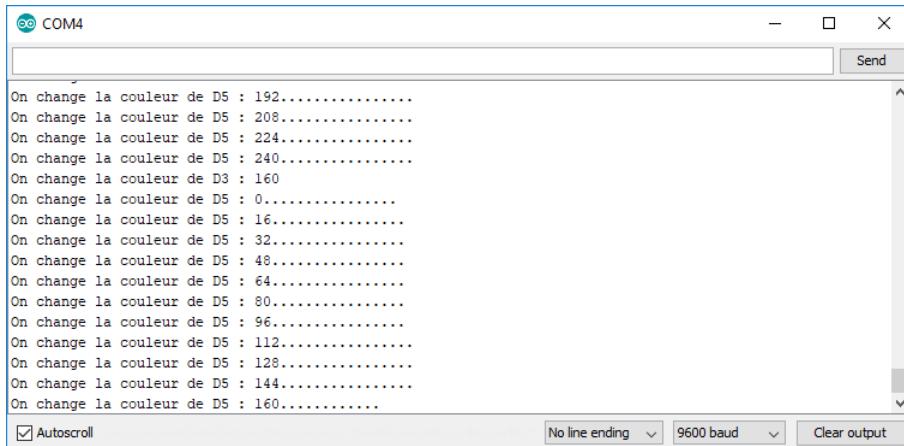
```

void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    Serial.begin(9600);
    pinMode(3, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
}

void loop() {
    for (int i=0; i <= 255; i+=16) {
        Serial.print("\nOn change la couleur de D3 : ");
        Serial.print(i);
        analogWrite(3, i);
        for (int j=0; j <= 255; j+=16) {
            Serial.print("\nOn change la couleur de D5 : ");
            Serial.print(j);
            analogWrite(5, j);
            for (int k=0; k <= 255; k+=16) {
                Serial.print(".");
                analogWrite(6, k);
                delay(50);
            }
        }
    }
}

```

Vous devriez voir apparaître les messages suivants :



Question : à quoi sert le début de la chaîne "\nOn change...

2^{ème} partie : Laboratoire en salle à Polytechnique

Un énoncé parmi les suivants sera assigné aléatoirement à chaque équipe.

Énoncé 1

Réalisez une application pour le contrôle d'une lumière à un passage pour piétons. En temps normal, la lumière est verte pour les voitures et rouge pour les piétons. Lorsqu'on pèse sur le bouton :

- ✓ La lumière passe au jaune pour les voitures pendant 10 secondes.
- ✓ Ensuite, elle passe au rouge et celle des piétons passe au vert.
- ✓ Après 20 secondes supplémentaires, la lumière des piétons passe au rouge.
- ✓ Finalement, 10 secondes plus tard, la lumière pour les voitures repasse au vert.
- ✓ Plus rien ne bouge pendant 30 secondes.

On est alors revenu à la situation de départ. Les détails techniques sont :

- On veut voir les DELs verte (sortie D2), jaunes (sortie D3) et rouge (sortie D4) s'allumer en séquence pour les voitures
- On veut voir d'autres DELs verte (sortie D5) et rouge (sortie D6) s'allumer en séquence pour les piétons
- Le bouton pour les piétons doit être réalisé avec le potentiomètre (entrée A0). On considère que le bouton est activé lorsque la tension fournie par le potentiomètre est supérieure à VCC/2

Énoncé 2

Réalisez un jeu de loterie électronique. Lorsqu'on pèse sur un bouton, six DELs monochromes s'allument en séquence (une après l'autre) très rapidement et le cycle recommence indéfiniment jusqu'à ce qu'on relâche le bouton. À ce moment-là, l'application mémorise quelle DEL était allumée. La séquence ralentit alors de plus en plus jusqu'à s'arrêter à la DEL mémorisée après un certain nombre de tours. Cette DEL restera allumée jusqu'à ce qu'on pèse sur le bouton à nouveau et que tout recommence.

Les détails techniques sont :

- Les DELs sont connectées aux ports D2 à D7 (dans l'ordre de la séquence)
- Le bouton doit être réalisé avec le potentiomètre (entrée A0). On considère que le bouton est activé lorsque la tension fournie par le potentiomètre est supérieure à VCC/2
- Le délai initial entre chaque transition est de 10ms
- Lorsque le bouton est relâché, le délai entre chaque transition augmente de 10ms à chaque transition, pendant 15 tours (90 transitions)

Énoncé 3

Réalisez un vu-mètre numérique. Cet appareil affiche l'intensité d'un signal audio. Le signal audio sera simulé par le potentiomètre. La tension d'entrée sera divisée en sept plages (0 à 6). La plage dans laquelle on se trouve doit être visualisée en allumant un nombre équivalent de DELs (dans l'ordre : 2 vertes, 2 jaunes et deux rouges). Par exemple, si on est dans la plage trois, il y aura deux DEL vertes et une DEL jaune allumées. Les détails techniques sont :

- Les DELs sont connectées aux ports D2 à D7 (dans l'ordre)
- Le potentiomètre utilise l'entrée A0
- Les plages sont divisées comme suit : 0-199 (plage 0), 200-399 (plage 1), 400-549 (plage 2), 550-699 (plage 3), 700-799 (plage 4), 800-899 (plage 5), 900-1023 (plage 6)