



# Robotique

Travaux Pratiques

Intervenants : Sophie Cavassila, Anne Laure Deman,  
Boussad Haroun, Hamid Ladjal, Francois Varray,  
28/06/2019

## Table des matières

1. Installation de l'environnement de développement Arduino.....	2
2. Les entrées et sorties numériques : Travail demandé.....	3
i) Sorties numériques - Projet : « Blink » .....	3
ii) Entrées numériques - Projet : « DIGITAL INPUTS » .....	14
iii) Sorties numériques : Ecran LCD .....	21
3. Communication série : Travail demandé .....	22
4. Les entrées analogiques : Travail demandé .....	26
5. Modulation de largeurs d'impulsions : Travail demandé .....	29
6. Exemple de projets à réaliser .....	30

## 1. Installation de l'environnement de développement Arduino

L'IDE Arduino (Arduino Integrated Development Environment (IDE)) est l'environnement de développement dédié aux cartes Arduino.

Le fichier d'installation de l'environnement de développement peut être téléchargé en ligne : <https://www.arduino.cc/en/Main/Software#download>

L'environnement de développement Arduino IDE est disponible pour les systèmes d'exploitation : Windows, Mac OS X et Linux.

**Le logiciel est installé sur tous les postes des salles de TP du bâtiment Nautibus**

Download the Arduino IDE



**ARDUINO 1.8.9**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
**Get**

**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

Release Notes  
Source Code  
Checksums (sha512)

Avec un support technique ici :

<https://www.arduino.cc/en/Guide/HomePage>

[https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation\\_arduino\\_ardublock\\_windows.pdf](https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:installation_arduino_ardublock_windows.pdf)

## 2. Les entrées et sorties numériques : Travail demandé

i) Sorties numériques - Projet : « Blink »

Ce premier projet permet de faire clignoter la led de la carte de développement connectée à la broche 13. La temporisation du clignotement est ajustable.

Dans le cadre de ce TP, nous allons exploiter les documents et codes librement proposés par Elegoo : (<http://www.elegoo.com/download/>) : Elegoo Mega 2560 The Most Complete Starter Kit,

Pour la mise en œuvre de ce projet, se référer au tutorial Elegoo “THE MOST COMPLETE STARTER KIT TUTORIAL FOR MEGA2560”: **Leçon 2 BLINK p32. Cette leçon est insérée à la page suivante.**



```
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(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the  
  voltage LOW  
  delay(1000); // wait for a second  
}
```

**Note:** Dans ce programme, LED\_BUILTIN est une constante qui est connue dans l’environnement de développement et qui est égale à 13, numéro de la broche à laquelle la led est connectée. Il est équivalent de déclarer une variable nommée led initialisée à 13 comme présenté dans le texte de la leçon 2.

## Leçon 2 Blink

### But de la leçon

Dans cette première leçon, vous allez prendre en main l'IDE et apprendre à faire clignoter la LED intégrée à la carte MEGA2560 R3.

### Matériel nécessaire:

(1) x Elegoo Mega2560 R3

### Principe

La carte MEGA2560 R3 possède deux rangées de connecteurs le long de ses extrémités qui sont utilisés pour brancher une vaste gamme de composants électroniques ou des cartes d'extensions (appelées shields) qui augmentent ses capacités.

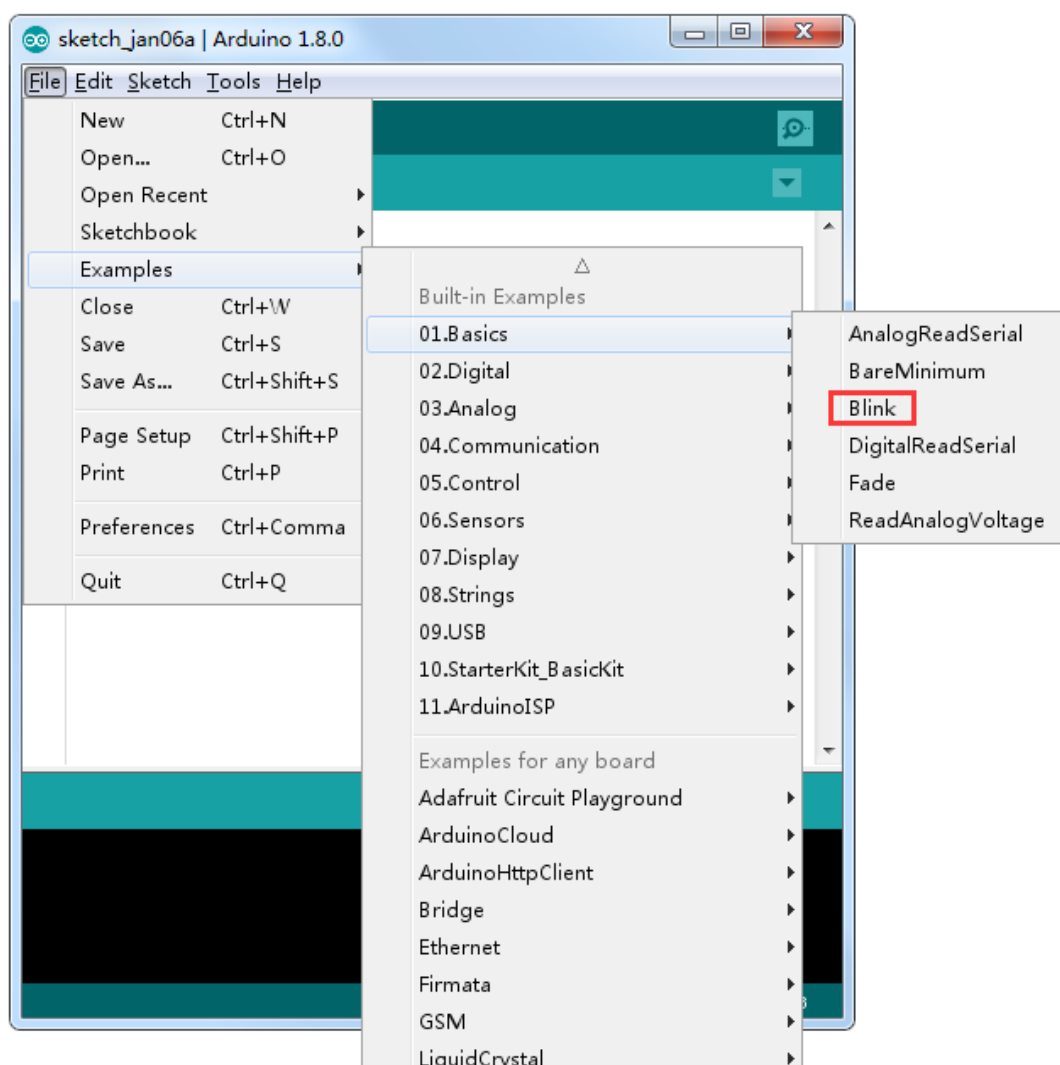
Elle est aussi équipée d'une LED intégrée qu'il est possible de commander au travers de vos programmes. Vous pouvez apercevoir cette LED sur l'image ci-dessous, elle est repérable grâce au « L » visible sur la carte.



Lorsque vous connectez votre carte pour la première fois, il est possible que la LED se mette à clignoter. C'est parce que les cartes sont fréquemment expédiées avec le programme « BLINK » préinstallé.

Dans cette leçon, nous allons reprogrammer la carte MEGA2560 R3 avec notre propre programme « BLINK », ce qui va nous permettre notamment de changer la fréquence de clignotement de la LED.

Ouvrez le sketch "BLINK" dans l'environnement de programmation via le menu "FICHIER/EXEMPLES/01.BASICS/Blink"



Voici ce que vous devez obtenir.

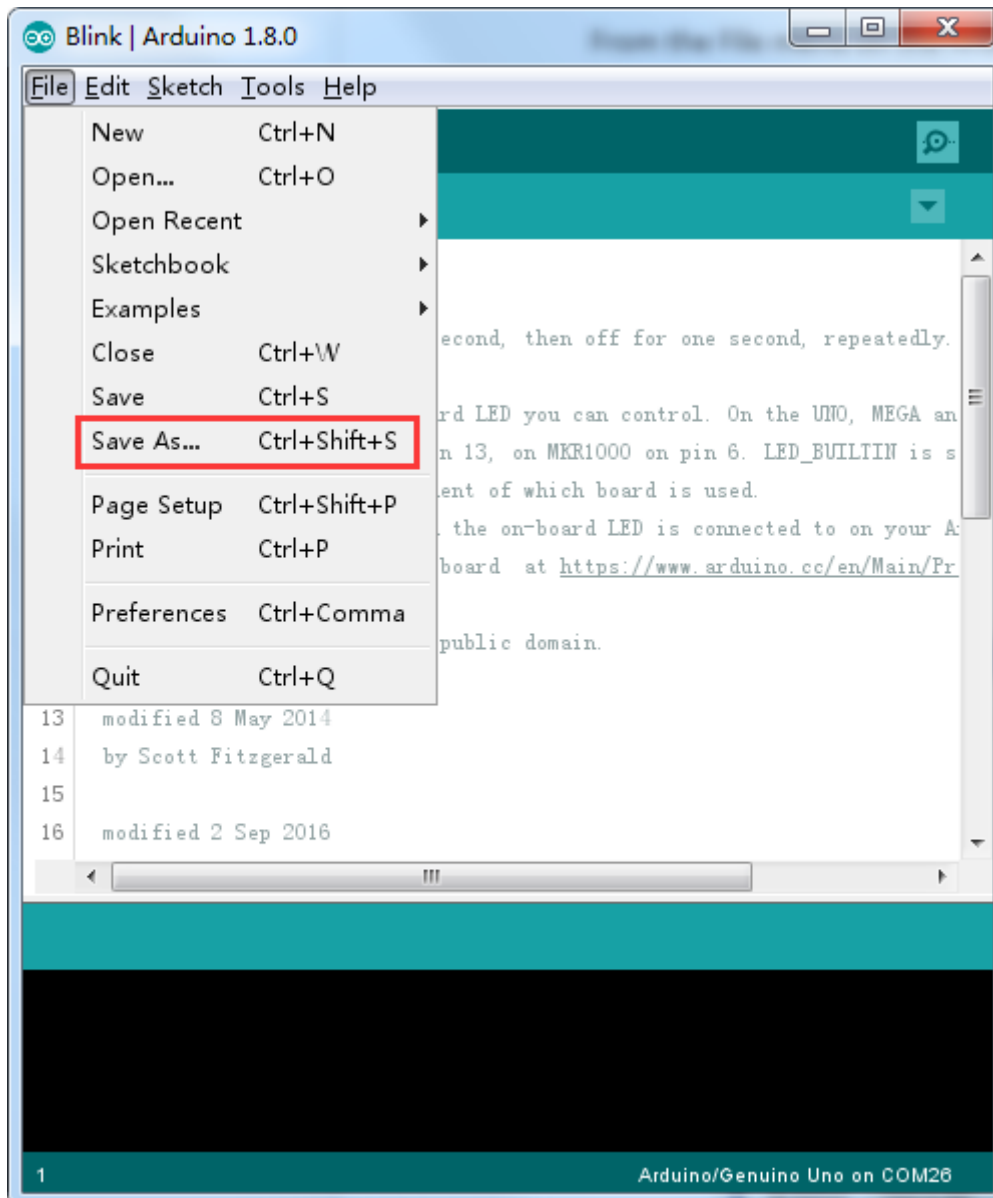
The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.8.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, an up arrow, a down arrow, and a magnifying glass. The main editor area displays the "Blink" sketch code, which is a multi-line comment block. The code is as follows:

```
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is s
7  the correct LED pin independent of which board is used.
8  If you want to know what pin the on-board LED is connected to on your A
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Pr
10
11  This example code is in the public domain.
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
```

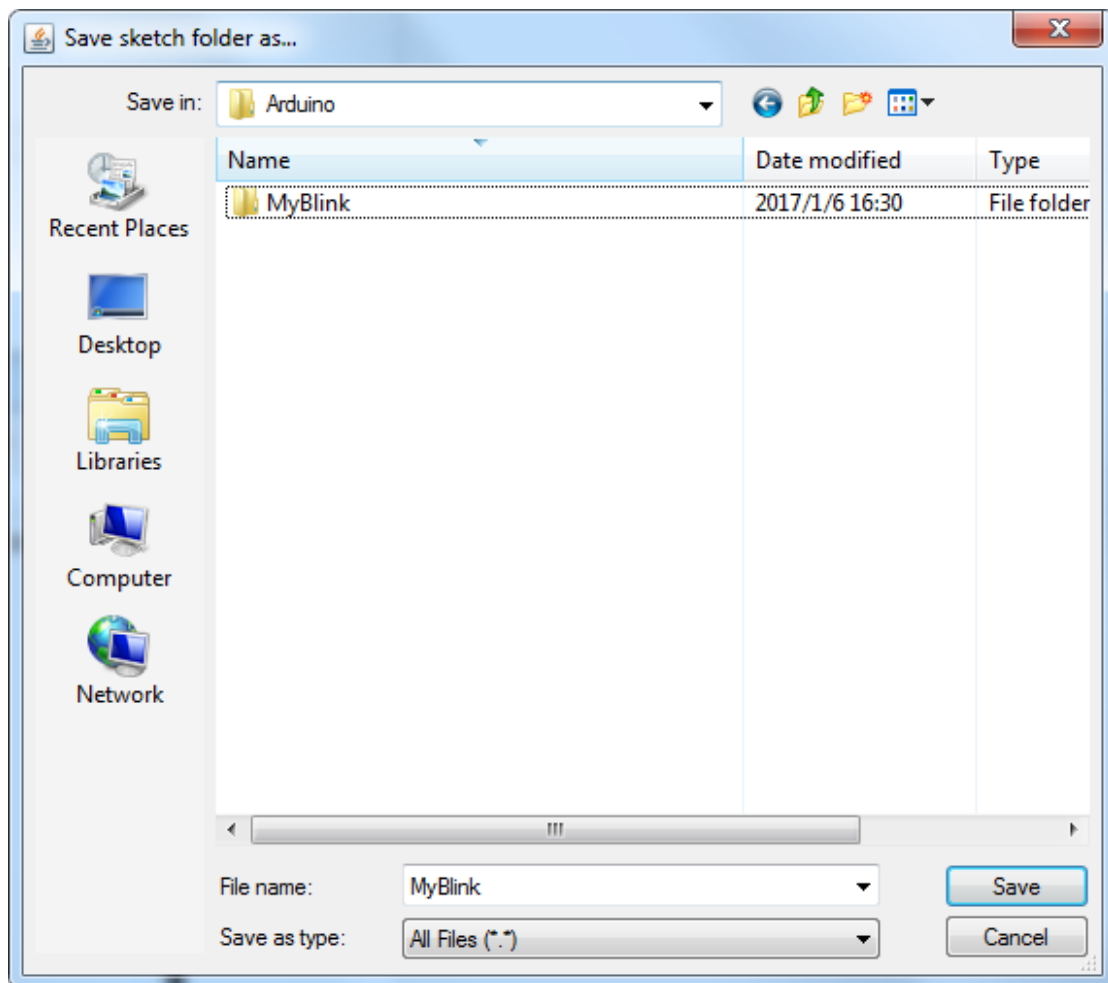
The status bar at the bottom of the IDE indicates "Arduino/Genuino Uno on COM28".

Comme vous allez apporter des modifications à ce sketch, la première étape consiste à l'enregistrer dans votre répertoire personnel, les sketches exemples étant en lecture seule.

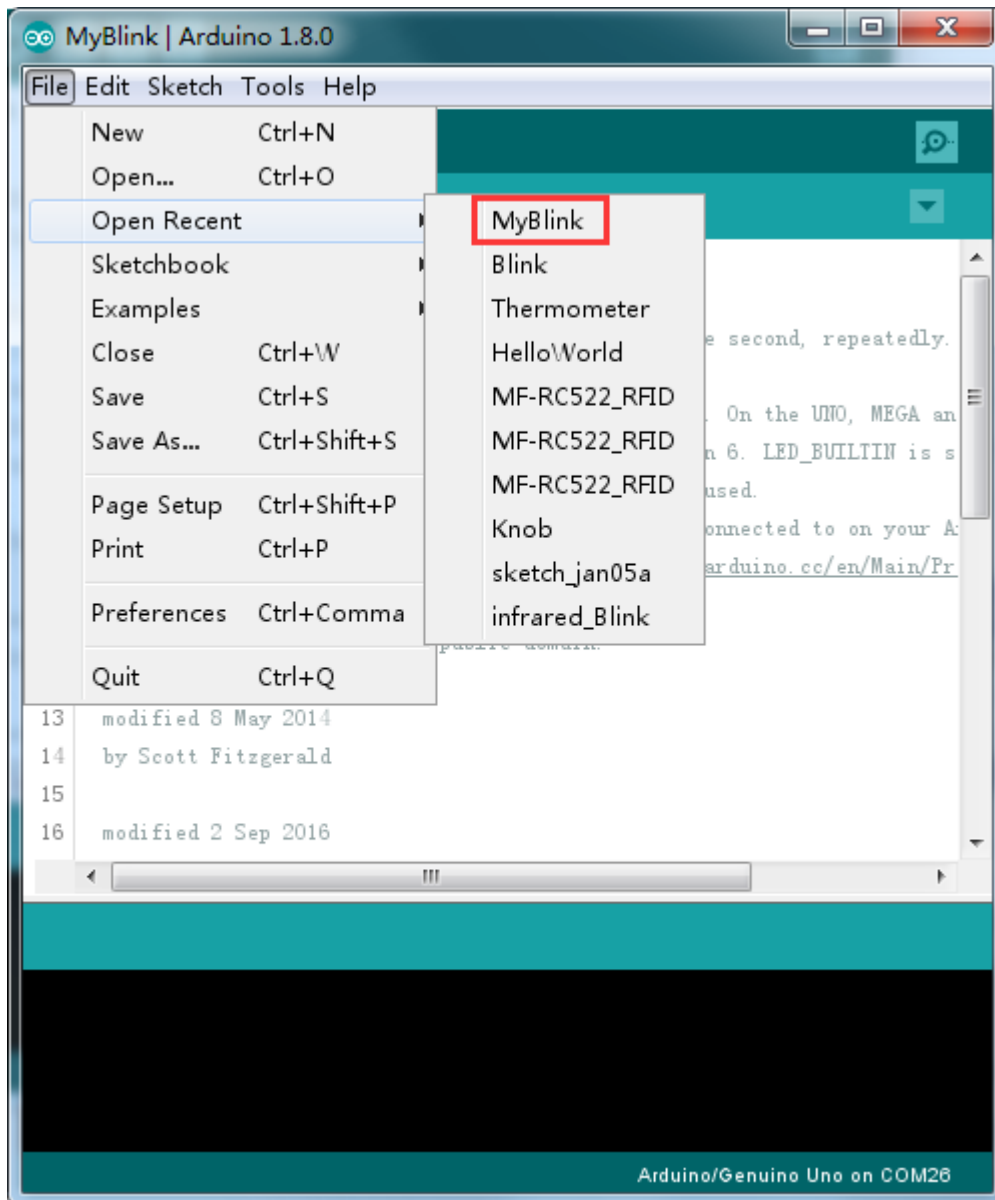
Utilisez le menu « enregistrer sous... » et définissez le répertoire et nom de fichier que vous souhaitez.





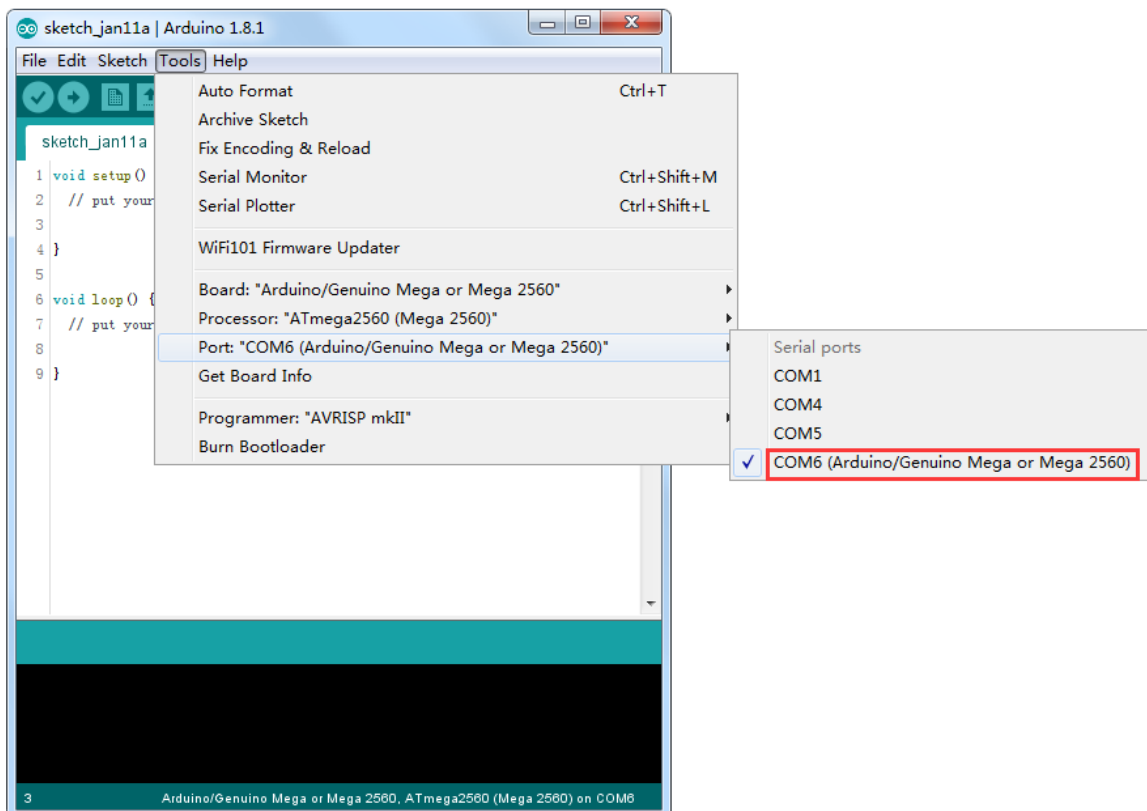
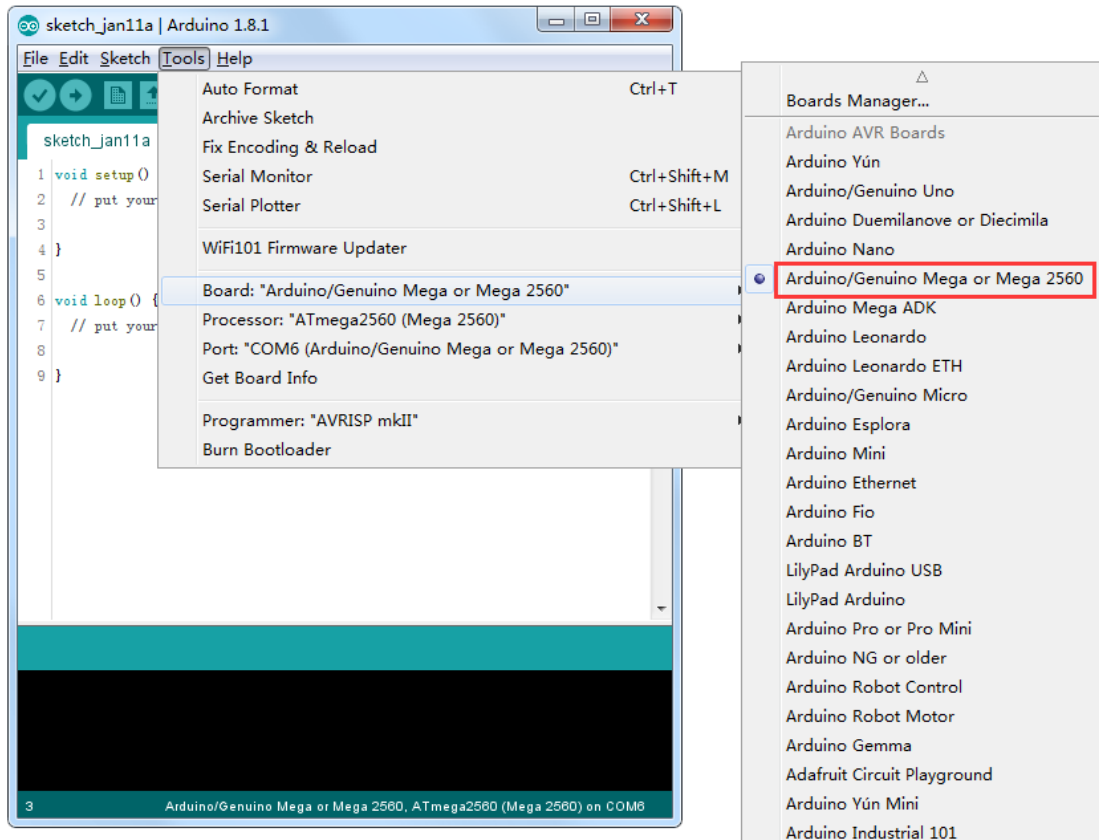


Vous venez d'enregistrer le fichier dans votre répertoire.



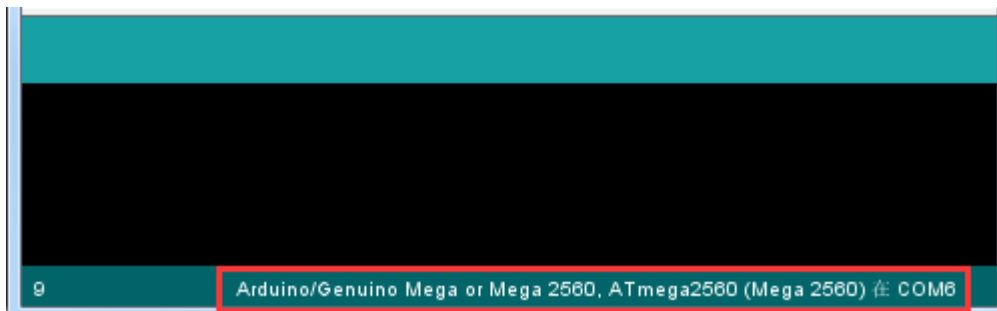
Vous pouvez accéder à vos fichiers récents de la même manière que la plupart de vos autres logiciels.

Il est temps maintenant de connecter la carte MEGA2560 R3 via le port USB de votre ordinateur et de vérifier la bonne reconnaissance de celle-ci.



**Note: le numéro de port COM ne sera pas nécessairement celui que vous pouvez apercevoir sur les captures d'écran. En effet, c'est votre ordinateur qui va affecter une référence au moment de la connexion et si vous possédez plusieurs cartes, chacune aura potentiellement un numéro différent.**

Une fois connecté, vous pouvez apercevoir ce petit texte en bas à droite de votre écran.



Cliquez sur le bouton avec la flèche pour déclencher le téléversement.



Vous pouvez constater que la première étape est la compilation (le logiciel vérifie l'intégrité du code).



Ensuite, le téléversement à proprement parler commence.



Enfin, le statut "terminé" s'affiche.

Un message vous renseigne sur la taille du programme et la quantité de mémoire utilisée sur la carte. Si votre carte n'est pas correctement connectée ou reconnue, vous obtiendrez l'écran suivant. Reportez-vous aux explications données dans les premières leçons pour veillez à avoir correctement installé les drivers.



Vous pouvez constater qu'une part importante du code est consacrée aux commentaires. Les commentaires sont importants ils permettent de bien comprendre le code, surtout quand celui-ci est complexe. Cela permet d'y revenir plus tard lorsqu'il est nécessaire de le faire évoluer.

Il existe deux façons de mettre du texte en commentaire. La première est de commencer la ligne par « // ». La deuxième manière permet de placer une portion de code/texte en commentaire (idéal lorsque l'on veut tester plusieurs alternatives par exemple). Débuter la zone commentaire par « /\* » et terminez celle-ci par « \*/ »

Dans le code « blink », la vraie première ligne d'instruction est la suivante:

```
int led = 13;
```

Cette instruction permet de créer une constant nommée "led" et de lui affecter la valeur numérique "13".

Le bloc suivant représente le code qui sera exécuté une unique fois lors de la mise sous tension de la carte.

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

Tous les sketches doivent obligatoirement avoir un bloc “setup” dans lequel vous pouvez placez autant d’instructions que nécessaires. Les instructions s’écrivent entre {}.

Dans « BLINK » la seule instruction consiste à affecter le pin 13 en tant que sortie (OUTPUT).

Il est aussi obligatoire dans un sketch d’avoir un bloc “loop”. Contrairement au premier bloc celui-ci s’exécutera en boucle tant que la carte sera sous tension.

```
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW  
    delay(1000);                // wait for a second  
}
```

Le bloc « loop » est constitué de 4 instructions. La première déclenche l’allumage de la led, la deuxième une attente de 1000ms, la troisième l’extinction de la led. La dernière instruction une attente de 1000ms.

Vous allez maintenant augmenter la fréquence de clignotement de la LED. Vous avez certainement deviné que le paramètre à changer est celui entre () sur les instructions « delay ». Téléversez le programme et observez.

```
30 // the loop function runs over and over again forever  
31 void loop() {  
32     digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the volt  
33     delay(500)                            // wait for a second  
34     digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo  
35     delay(500)                            // wait for a second  
36 }
```

## ii) Entrées numériques - Projet : « DIGITAL INPUTS »

Ce projet permet de lire l'état de 2 boutons-poussoirs connectés aux broches 8 et 9 pour allumer ou éteindre une led connectée à la broche 5.

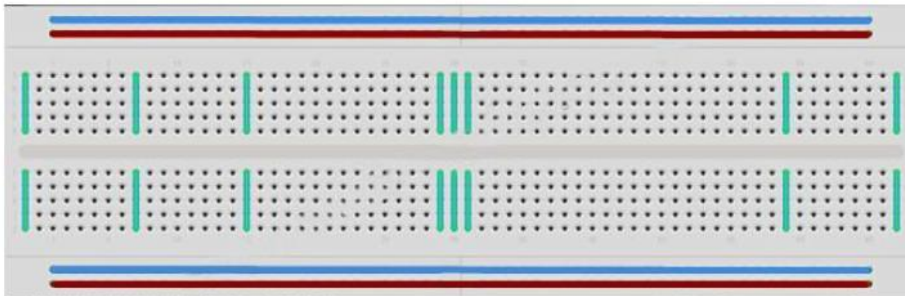
Dans le cadre de ce TP, nous allons exploiter les documents et codes librement proposés par Elegoo : (<http://www.elegoo.com/download/>) : Elegoo Mega 2560 The Most Complete Starter Kit :

### Leçon 5 DIGITAL INPUTS p58

## Précision sur l'utilisation d'une « planche de prototypage »

Une planche prototype vous permet de réaliser des circuits très rapidement, sans avoir besoin de réaliser de soudures.

Exemple :



Il existe une grande variété de planches prototypes. La plus simple est une grille de trous dans un bloc de plastique. A l'intérieur se trouvent des lames métal permettant la connexion électrique entre les différents trous d'une même ligne. Mettre les branches de deux composants sur une même ligne les "assemble" électriquement. La ligne creusée au centre de la plaque symbolise une rupture électrique entre la partie haute et la partie basse. Cela signifie aussi que vous pouvez positionner une puce à cheval entre haut et bas. Certaines planches prototypes ont aussi deux lignes horizontales en haut et en bas. On les utilise généralement pour créer une ligne d'alimentation +5V et masse.

Attention tout de même, les planches prototypes ont comme limite d'utilisation la qualité des connexions qui ne valent pas une soudure et peuvent entraîner parfois des dysfonctionnements.

#### LED:

Les leds font de parfaits indicateurs lumineux. Elles consomment peu de courant et ont une très bonne durée de vie.

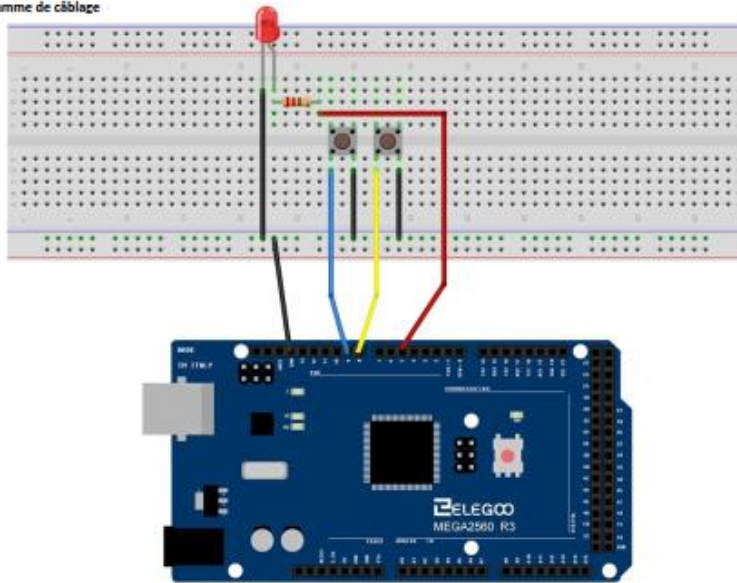
Dans cette leçon, vous allez utiliser certainement le type de leds le plus commun, la led 5mm. Il en existe d'autres de 3 à 10mm.

**Attention: il n'est pas possible de brancher directement une led sur une batterie car l'intensité de courant est trop forte.**



Attention aussi au sens de branchement de la led. La patte la plus longue doit être connectée à la borne + et la patte courte à la borne -.

Diagramme de câblage



```
int ledPin = 5;
int buttonApin = 9;
int buttonBpin = 8;

byte leds = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonApin, INPUT_PULLUP);
  pinMode(buttonBpin, INPUT_PULLUP);
}

void loop()
{
  if (digitalRead(buttonApin) == LOW)
  {
    digitalWrite(ledPin, HIGH);
  }
  if (digitalRead(buttonBpin) == LOW)
  {
    digitalWrite(ledPin, LOW);
  }
}
```



## Leçon 5 Digital Inputs

### But de la leçon

Dans cette leçon, vous allez apprendre à utiliser les entrées digitales pour allumer/éteindre une led.

Presser un premier bouton allumera la led, presser un autre bouton l'éteindra.

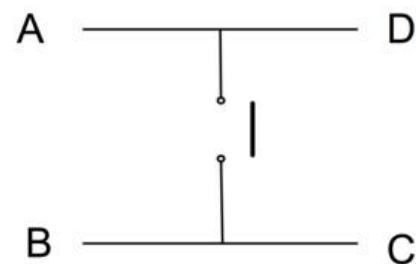
### Matériel nécessaire:

- (1) x Elegoo Mega2560 R3
- (1) x Planche prototype
- (1) x LED rouge 5mm
- (1) x résistance 220 ohms
- (2) x boutons poussoirs
- (7) x Câbles mâle-mâle

### Présentation du composant

#### Boutons poussoirs:

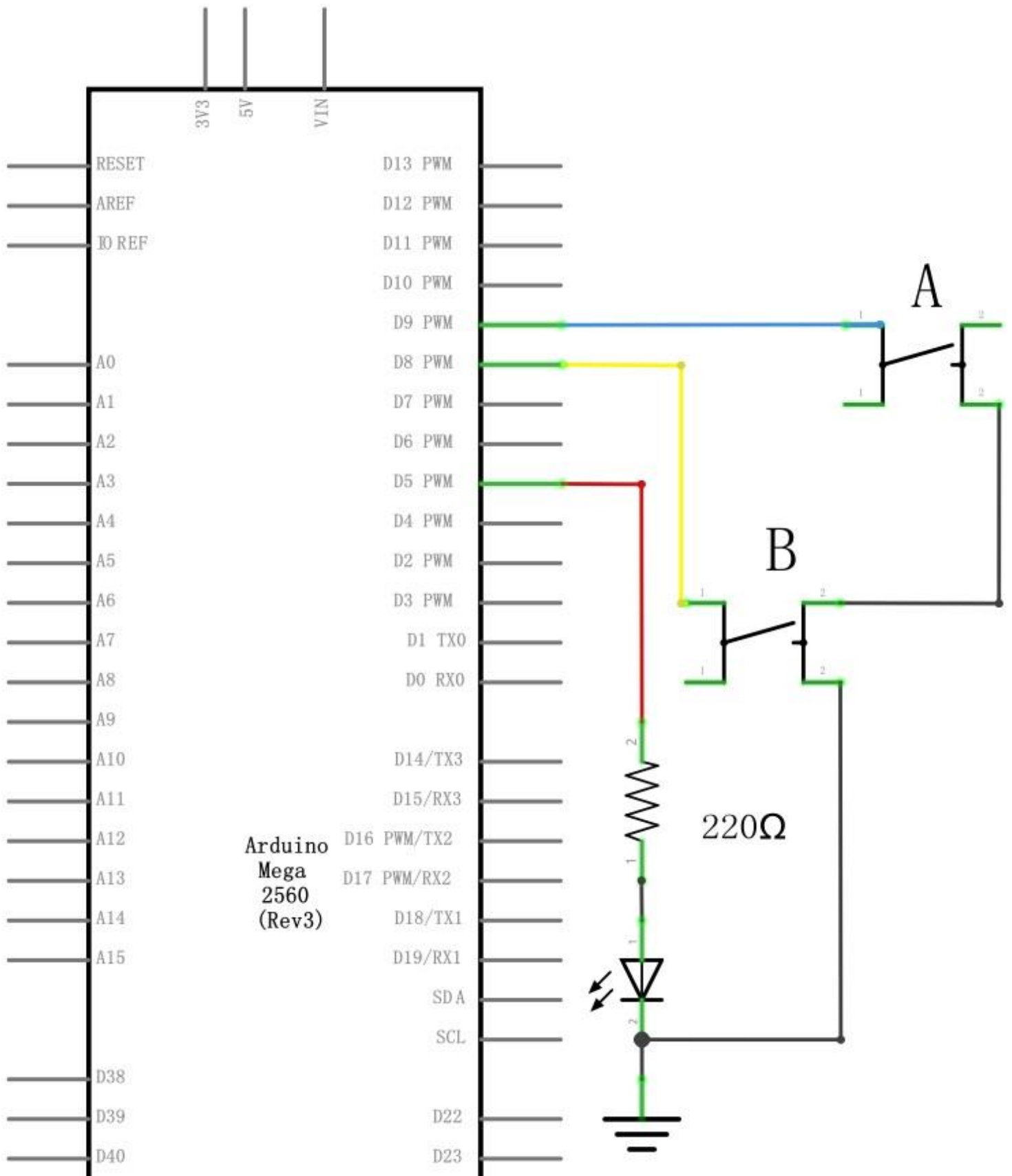
Les boutons poussoirs sont des composants très simples. Lorsque vous pressez le bouton, un contact électrique se fait et laisse passer le courant. Les boutons poussoirs utilisés ont 4 pattes, ce qui peut créer une certaine confusion.



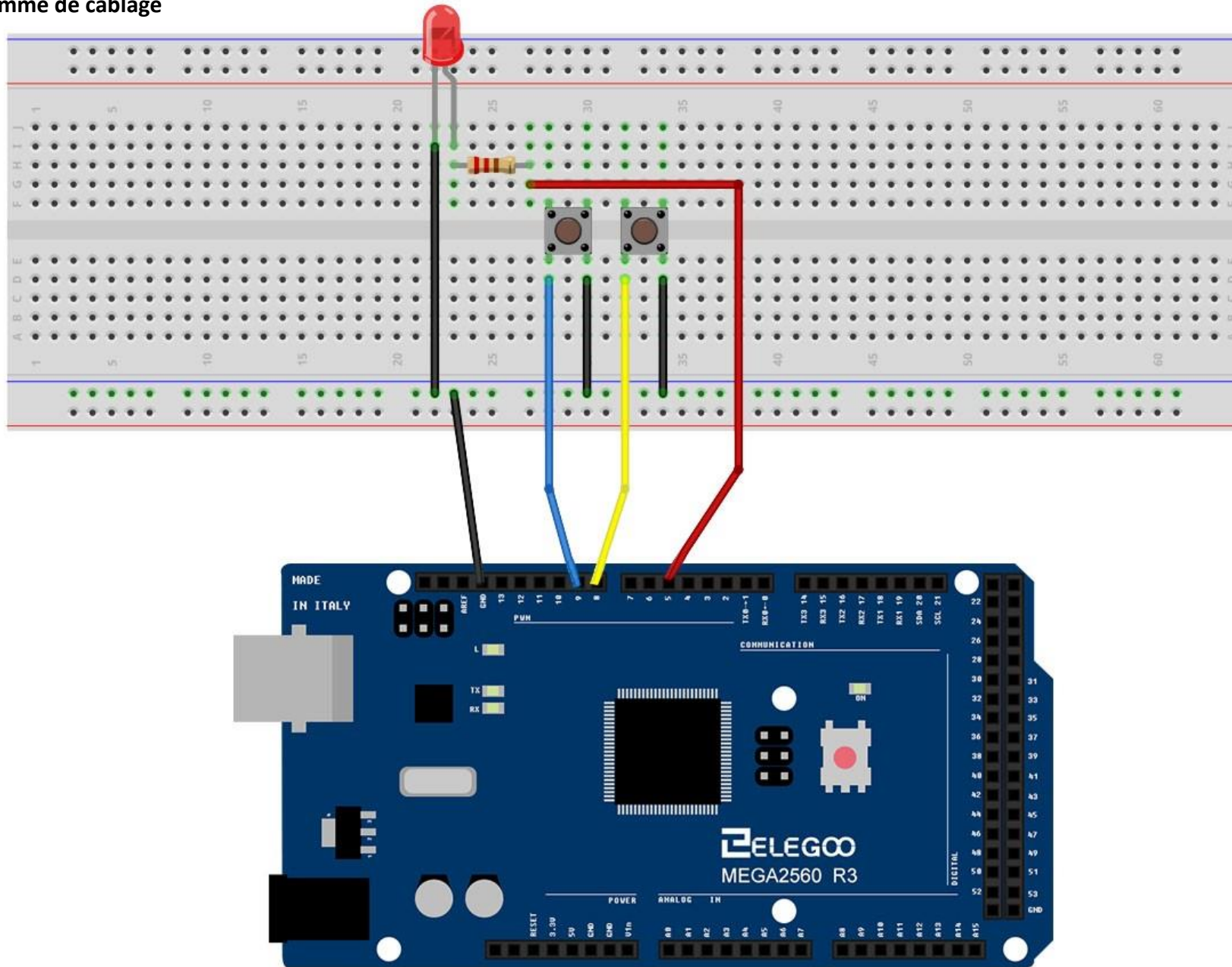
En fait, il n'y a bien que 2 connexions électrique. A et D sont connectées ensemble et B et C aussi. Presser le bouton permet au courant de lier électriquement A-D avec B-C.

# Connexion

## Schéma de câblage



# Diagramme de câblage



## Code

Après avoir réalisé le câblage, ouvrez le sketch “Leçon 5 Digital Inputs” puis téléversez le code sur la carte MEGA2560 R3 comme présenté lors de la leçon 2.

**Presser le bouton de gauche allumera la led, presser le bouton de droite l'éteindra.**

La première partie du sketch est consacrée à la définition des 3 variables pour les 3 pins qui seront nécessaires au fonctionnement du montage.

Dans le bloc « setup », les pins sont affectées en tant qu'entrées ou sorties.

'ledPin' est défini en tant que sortie (OUTPUT), 'buttonApin' et 'buttonBpin' en tant qu'entrées (INPUT).

```
pinMode(buttonApin, INPUT_PULLUP);
```

```
pinMode(buttonBpin, INPUT_PULLUP);
```

Petite particularité, les entrées sont définies en tant qu'entrées avec résistance de PULLUP. Pourquoi cette subtilité ? Comme vous pouvez le constater sur le schéma de câblage, le fait de presser un bouton met la pin associée à la masse. Mais lorsque le bouton n'est pas pressé, il peut subsister des signaux parasites qui peuvent être interprétés par la carte comme une mise à la masse alors que le bouton n'est pas pressé. Pour éviter ces parasites, on utilise la technique dite de « la résistance de pullup » qui permet de forcer un état haut via une résistance connectée à une borne +. La carte MEGA2560 est capable de simuler ce branchement, pour cela, il suffit de déclarer la pin en INPUT\_PULLUP au lieu de faire une déclaration INPUT simple.

Le bloc loop prend ensuite la mesure de la position de chaque bouton:

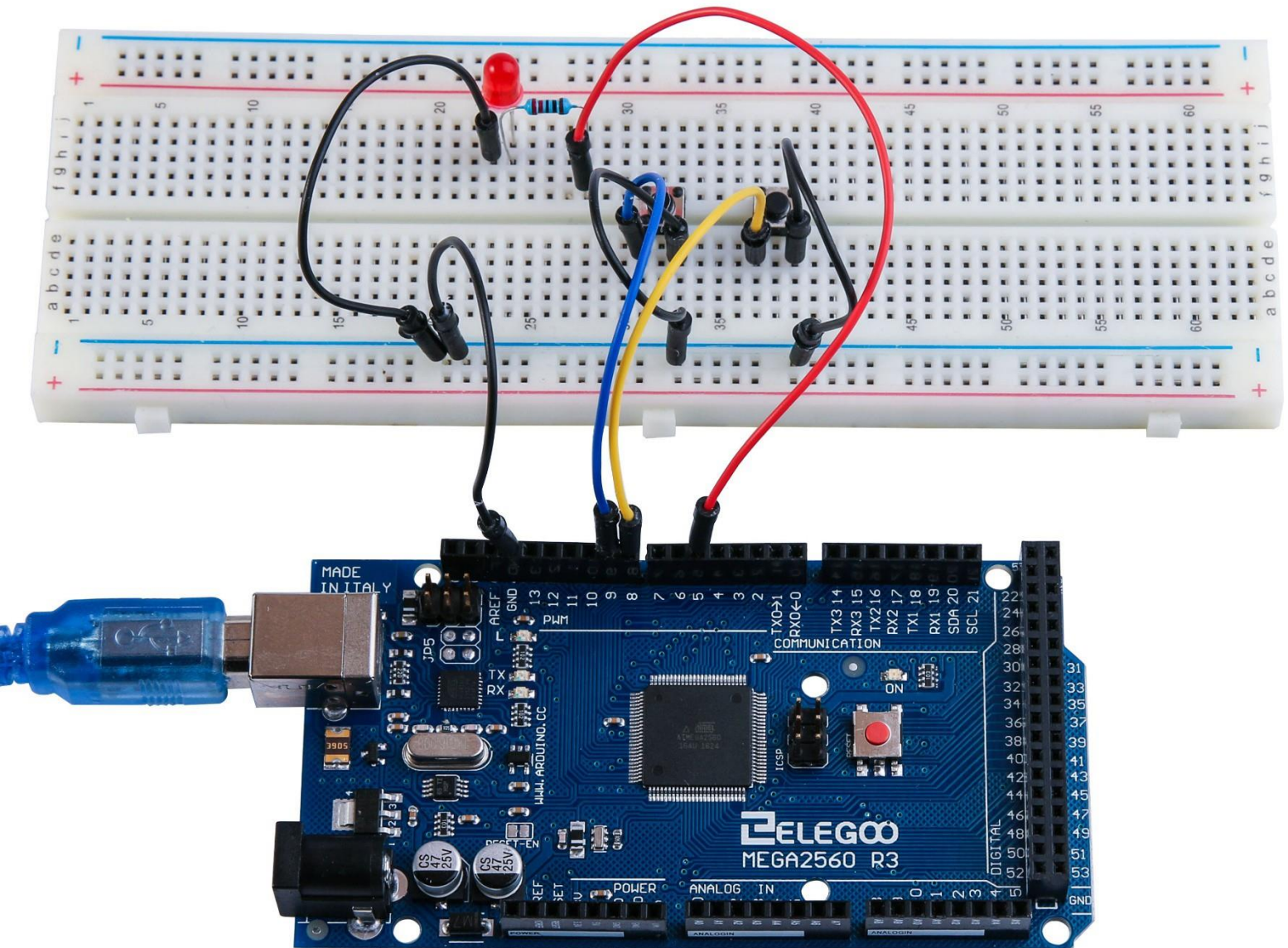
```
void loop()
{
  if (digitalRead(buttonApin) == LOW)
  {
    digitalWrite(ledPin, HIGH);
  }
  if (digitalRead(buttonBpin) == LOW)
  {
```

```
digitalWrite(ledPin, LOW);  
}  
}
```

On regarde si le bouton A est pressé (LOW). Si oui, on allume la led, si non, on ne fait rien.

On regarde ensuite si le bouton B est pressé. Si oui on éteint la led, si non, on ne fait rien.

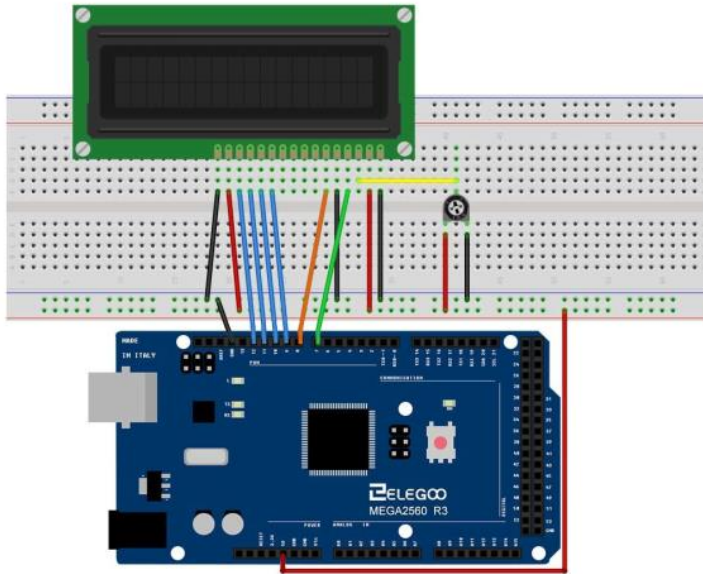
## Illustration



iii) Sorties numériques : Ecran LCD

Ce projet permet d'écrire des messages et des données numériques sur un écran LCD

Dans le cadre de ce TP, nous allons exploiter les documents et codes librement proposés par Elegoo : (<http://www.elegoo.com/download/>) : **Elegoo Mega 2560 The Most Complete Starter Kit** : Leçon 22 LCD DISPLAY p149



Le projet fait appel à la bibliothèque de fonctions LiquidCrystal.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Hello, World!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

### 3. Communication série : Travail demandé

Le microcontrôleur est ici configuré pour émettre et recevoir des informations via sa liaison série. Les données échangées sont visibles dans la fenêtre moniteur série.

Exécuter le programme mis à disposition ci-dessous qui permet d'éclairer la led connectée à la broche 13 si l'information reçue sur la liaison série est égale au caractère « 1 » et de l'éteindre si l'information reçue correspond au caractère « 0 ». L'utilisation du moniteur série est décrite page suivante (ou Elegoo Mega 2560 The Most Complete Starter Kit : page 29). La liaison série sera configurée avec le format par défaut et une vitesse de transmission de 9600bps.

Modifier ce programme de façon à ce que la led change d'état selon la réception des caractères suivants :

- A : éclairer la led
- Z : éteint la led

```
#define led 13
int recue = 0; // pour les données reçues

void setup() {
  Serial.begin(9600); // configure la liaison série avec une vitesse de 9600 bps
  pinMode(led, OUTPUT); //configure la broche 13 en sortie
}

void loop() {
  if (Serial.available() > 0) // Test si une donnée a été reçue
  {
    //lecture des données reçues:
    recue = Serial.read();

    if (recue == '1') {
      digitalWrite(led,HIGH); //positionne la broche 13 à l'état haut
      Serial.print("Eclairé "); //envoie un message via la liaison série
    }

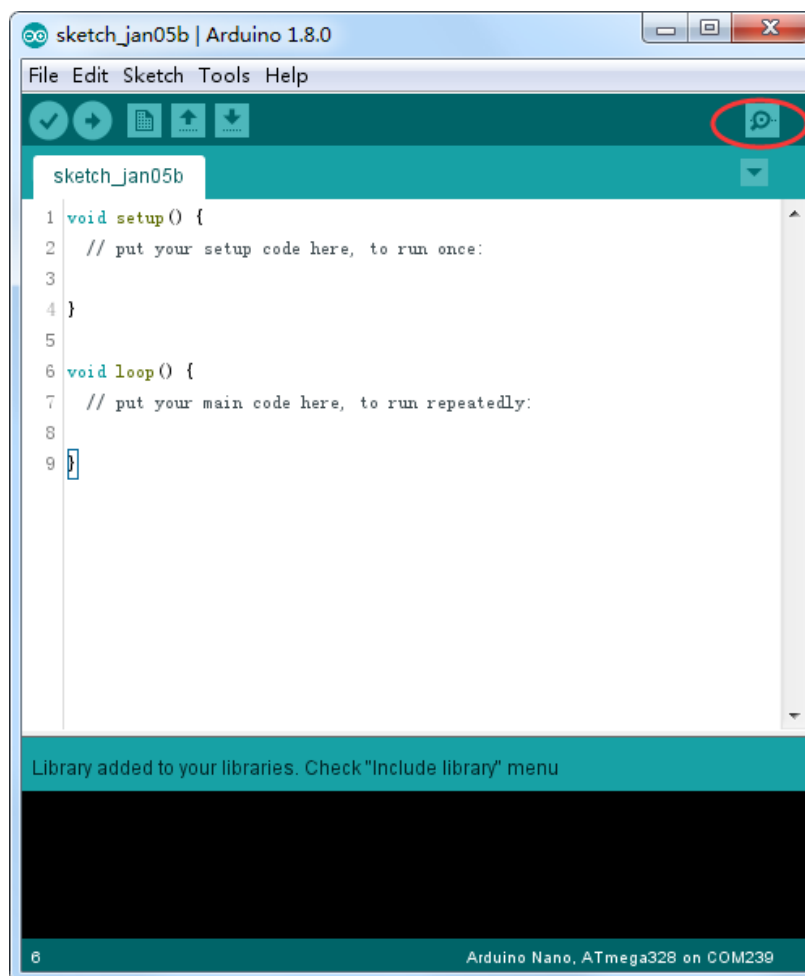
    if (recue == '0') {
      digitalWrite(led,LOW); //positionne la broche 13 à l'état bas
      Serial.print("Eteint "); //envoie un message via la liaison série
    }
  }
}
```

## Utilisation du moniteur série (Windows, Mac, Linux)

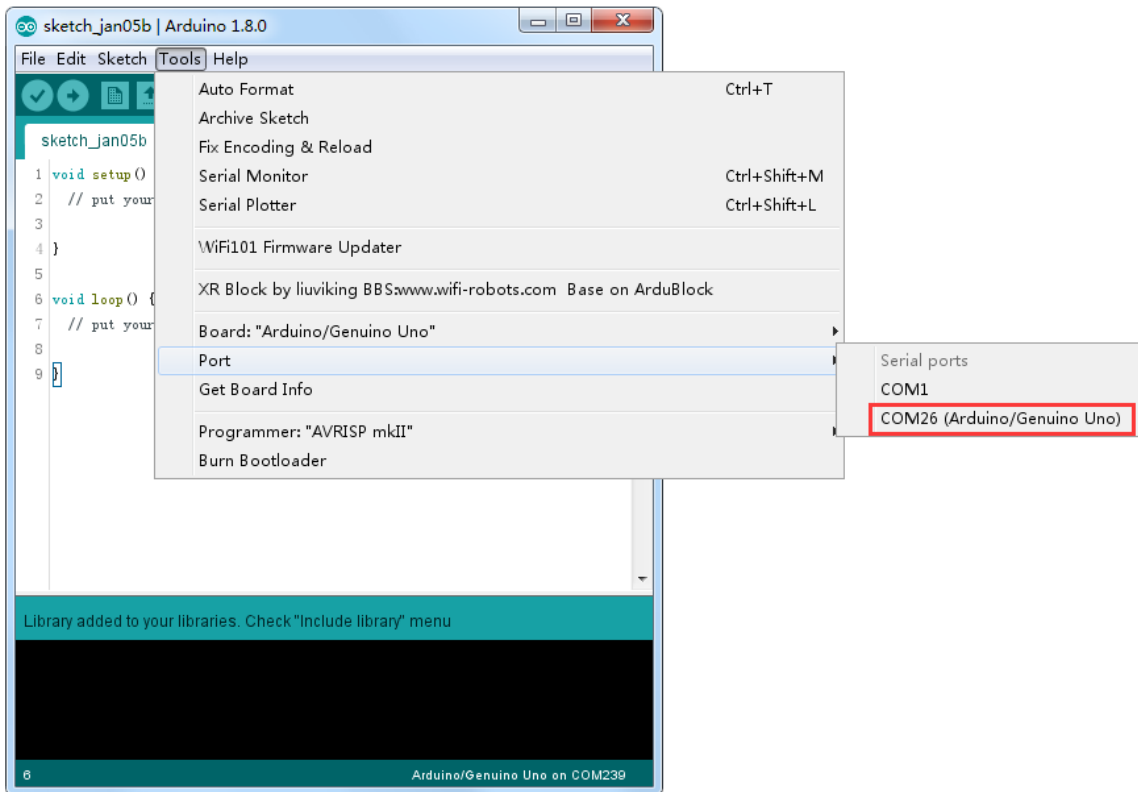
Le moniteur série est une petite fenêtre très utile qui va vous permettre d'interagir en temps réel avec la carte MEGA2560. Vous pouvez lui envoyer des informations et elle pourra à son tour faire de même. Vous verrez dans les différentes leçons que le recours au moniteur série est très fréquent.

### Se Connecter

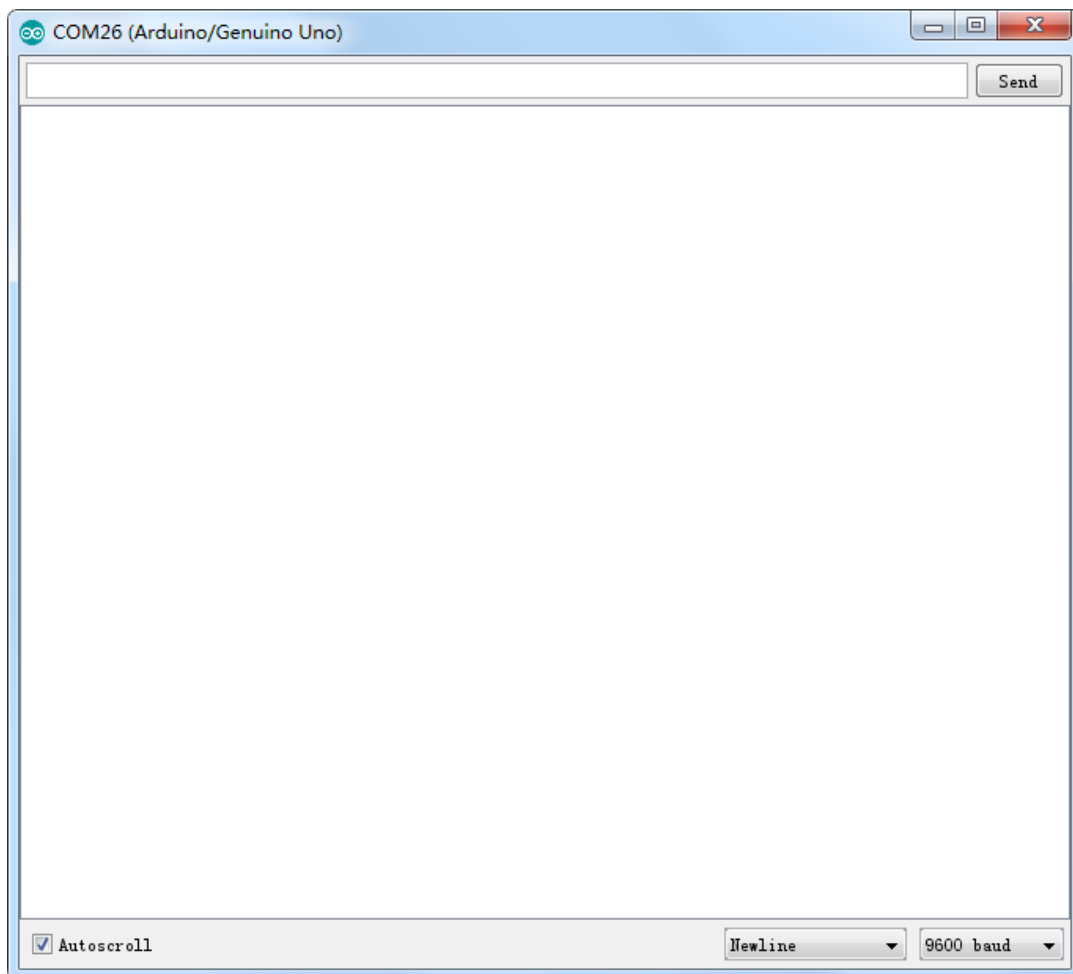
Pour l'ouvrir, c'est très simple, il suffit de cliquer sur le petit icône entouré en rouge.





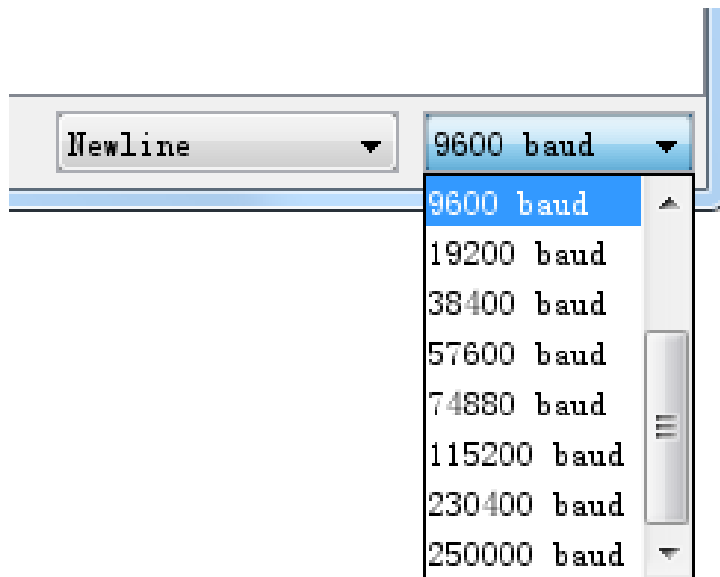


A l'ouverture, vous devez voir cette fenêtre:

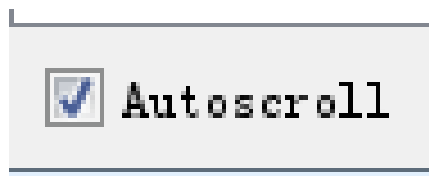


## Settings

Vous pouvez définir le taux de transfert de la connexion de la manière suivante



Et vous pouvez avoir un défilement automatique de l'écran comme suit



### Avantages

Le Serial Monitor est un excellent moyen d'établir rapidement une connexion série avec votre Arduino. Si vous travaillez déjà dans l'IDE Arduino, il n'est vraiment pas nécessaire d'ouvrir un terminal distinct pour afficher les données.

### Les inconvénients

Le manque de paramètres laisse beaucoup à désirer dans le Serial Monitor, et, pour les communications en série avancées, il se peut que ce ne soit pas le tour.

## 4. Les entrées analogiques : Travail demandé

Dans le cadre de ce TP, nous allons nous appuyer sur le projet proposé par Arduino IDE pour illustrer la conversion analogique numérique : Exemples->Basics : ReadAnalogVoltage.

Ce programme réalise la conversion analogique numérique de la tension connectée sur la broche A0 du convertisseur et affiche dans la fenêtre Moniteur Série la valeur de la tension correspondante.

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
}
```

Il s'agit de mettre en oeuvre ce programme pour mesure les tensions délivrer au choix par :

- Un joystick (cf Elegoo Mega 2560 The Most Complete Starter Kit [p96](#))

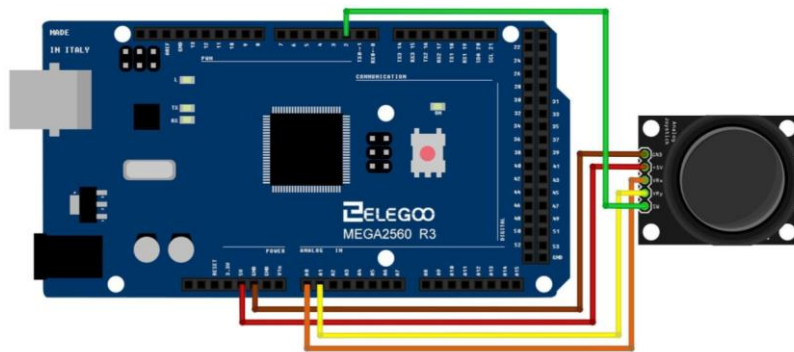
### Joystick

Le module a 5 pins: VCC, Ground, X, Y, Key. Les noms affectés aux pins peuvent légèrement varier en fonction de la provenance des modules. Le joystick est analogique, ce qui permet une lecture précise de la position et offre beaucoup plus de possibilité qu'un simple joystick tactile. En plus, il est possible d'appuyer sur le joystick pour faire une fonction « select ».

Pour le câblage, la connexion du signal de validation à la broche 2 du microcontrôleur n'est pas nécessaire car inutilisée par le programme.

Axe X connecté à la broche A0

Axe Y connecté à la broche A1 (pas pris en compte par le programme proposé ci-dessus).



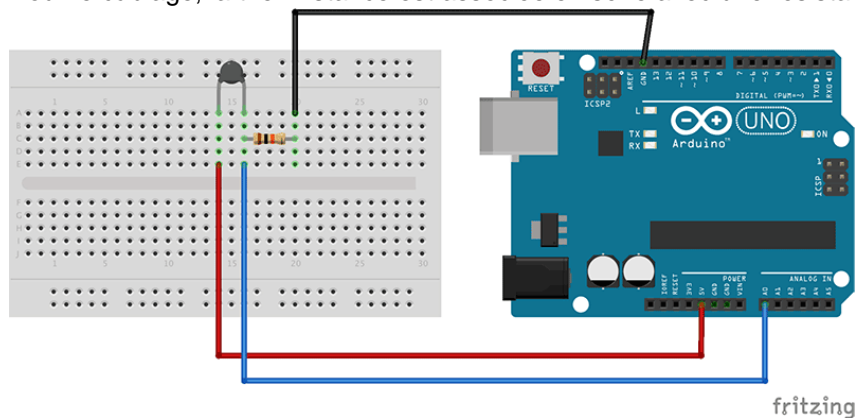
- Une thermistance (cf Elegoo Mega 2560 The Most Complete Starter Kit p154 pour la connexion)

### Thermistance

Une thermistance est une “résistance thermique” : sa valeur varie en fonction de la température. Une résistance standard voit aussi sa température varier, mais très faiblement, une thermistance est faite pour cette variation soit importante et donc facile à mesurer. (100 ohms par degré)

Il en existe deux types : NTC (negative temperature coefficient) et PTC (positive temperature coefficient). En général, on utilise des NTC.

Pour le câblage, la thermistance est associée en série avec une résistance de 10kOhm.



- Une photorésistance (cf Elegoo Mega 2560 The Most Complete Starter Kit p171 pour la connexion)

### Présentation du composant

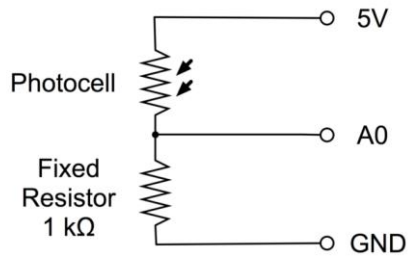
#### PHOTOCELL:

La photocell utilisée est appelée “résistance dépendante de la lumière” ou light dependent resistor (LDR). Comme son nom l’indique la valeur de la résistance dépend de la quantité de lumière reçue par le composant.

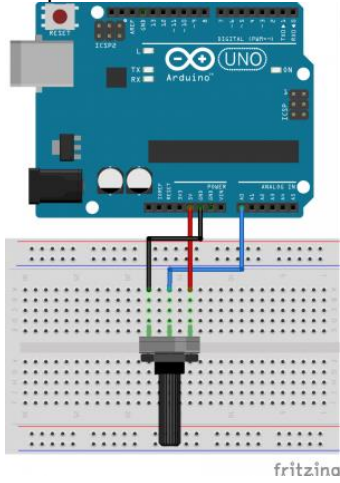
Cette résistance prend une valeur de 50 kΩ en obscurité et varie jusqu’à 500 Ω en pleine lumière. Pour convertir cette variation de valeur en variation de courant, il faut bien entendu alimenter la résistance.



Pour le câblage, la photorésistance est associée en série avec une résistance de 1kOhm.



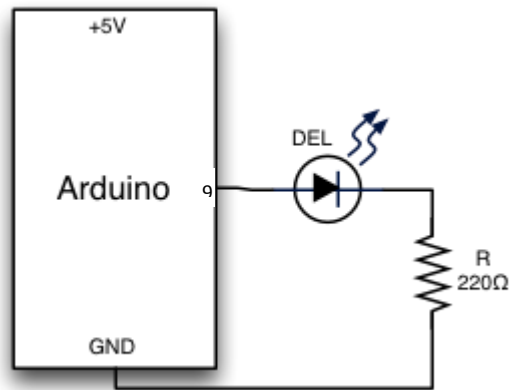
- Un potentiomètre en connectant le point milieu à la broche A0.



## 5. Modulation de largeurs d'impulsions : Travail demandé

Dans le cadre de ce TP, nous allons nous appuyer sur le projet proposé par Arduino IDE pour illustrer la conversion analogique numérique : Exemples->Basics-> **Fade**.

Ce programme permet de faire varier la luminosité de la diode en générant un signal modulé en largeur d'impulsion sur la broche 9 du microcontrôleur.



```
int led = 9;           // the pin that the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Pour aller plus loin il est possible de mettre en œuvre la **leçon 29 Elegoo Mega 2560 The Most Complete Starter Kit p187**. Il s'agit de piloter la vitesse de rotation d'un moteur en générant un signal PWM.

## 6. Exemple de projets à réaliser

### Commande des déplacements d'un robot et de son système lumineux

- Il s'agit de proposer un projet permettant la réalisation d'un robot piloté par une application android communicant en Bluetooth ou une télécommande infra-rouge. L'application permettra le contrôle des mouvements du robot ainsi que de ses signaux lumineux.
- Le robot peut être équipé d'un détecteur d'obstacle qui peut renvoyer des informations au système de pilotage.
- De nombreux éléments de présentation de capteurs et actionneurs sont à disposition (<http://www.elegoo.com/download/>) et libres pour le téléchargement : **Elegoo Mega 2560 The Most Complete Starter Kit, Elegoo Smart Robot Car Kit V3.0**

### Domotique : Contrôle de l'intensité de l'éclairage

- En mode automatique : l'intensité lumineuse est fixée en fonction de la luminosité ambiante
- En mode manuel : l'intensité lumineuse est fixée via un variateur de type potentiomètre.
- Le mode automatique ou manuel est choisi par exemple par un interrupteur. Le système d'éclairage est une matrice de leds.
- De nombreux éléments de présentation de capteurs et actionneurs sont à disposition (<http://www.elegoo.com/download/>) et libres pour le téléchargement : **Elegoo Mega 2560 The Most Complete Starter Kit, Elegoo Smart Robot Car Kit V3.0**

### Domotique : Contrôle de la vitesse de rotation d'un ventilateur

- En mode automatique : la vitesse de rotation du moteur du ventilateur est fixée en fonction de la température ambiante
- En mode manuel : la vitesse de rotation du moteur du ventilateur est fixée par un potentiomètre
- Le mode automatique ou manuel est choisi par exemple par un interrupteur.
- De nombreux éléments de présentation de capteurs et actionneurs sont à disposition (<http://www.elegoo.com/download/>) et libres pour le téléchargement : **Elegoo Mega 2560 The Most Complete Starter Kit, Elegoo Smart Robot Car Kit V3.0**

### Alarme : Système de contrôle des accès

- Les accès d'une pièce sont sécurisés. Le système de sécurité est activé par un code saisi sur un clavier matriciel pu par une carte rfid Les intrusions peuvent détectées par plusieurs capteurs de type capteur de mouvement, capteur de distance, capteur de son. Les alertes sont données par des dispositifs lumineux et sonores.
- De nombreux éléments de présentation de capteurs et actionneurs sont à disposition (<http://www.elegoo.com/download/>) et libres pour le téléchargement : **Elegoo Mega 2560 The Most Complete Starter Kit, Elegoo Smart Robot Car Kit V3.0**