

Formation Arduino

Hainaut Patrick 2021

But de cette présentation

- Les microcontrôleurs représentent la frontière entre l'électronique et l'informatique
- Ils permettent de créer assez facilement des systèmes "intelligents", autonomes, interactifs,
- Ils envahissent notre quotidien, notamment au travers de l'internet des objets (IoT Internet of Things)
- Un informaticien se doit d'être sensibilisé au monde des microcontrôleurs

©Hainaut P. 2021 - www.coursonline.be

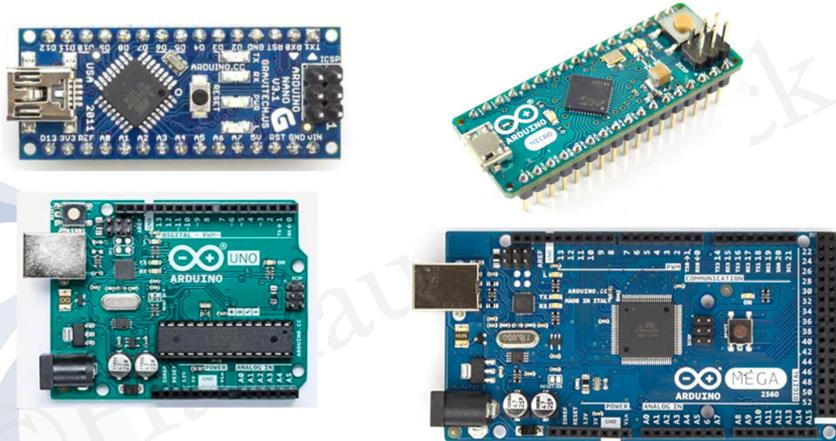
2

Introduction

- L'arduino est un des microcontrôleurs les plus populaires avec le raspberry pi, c'est pourquoi nous allons nous concentrer sur celui-ci
- L'arduino va nous permettre de nous initier à la programmation de façon progressive tout en rendant cette programmation "visuelle" et ludique

1. L'ARDUINO UNO

Quelques modèles populaires d'Arduino



- Vous pouvez la liste complète des cartes sur <https://www.arduino.cc/en/Main/Products>

©Hainaut P. 2021 - www.coursonline.be

5

Arduino Uno

- C'est un des modèles les plus utilisés et les moins chers
- Situé entre l'arduino Mega et les arduino Micro et Nano, l'arduino Uno est un bon compromis



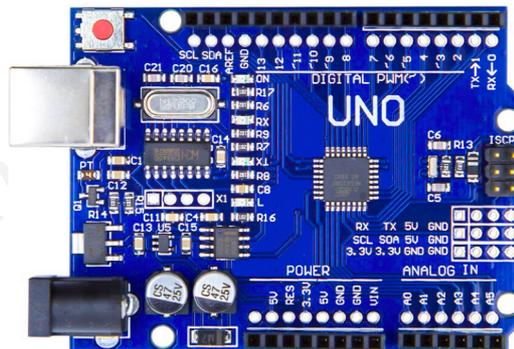
- La puce du microcontrôleur peut revêtir deux formes sans que cela n'affecte les performances

©Hainaut P. 2021 - www.coursonline.be

6

Disponibilité en chine

- Pour des tests et des montages peu critiques, une version chinoise à quelques euros suffit (disponible sur aliexpress)
- La version actuelle est l'arduino Uno R3



©Hainaut P. 2021 - www.coursonline.be

7

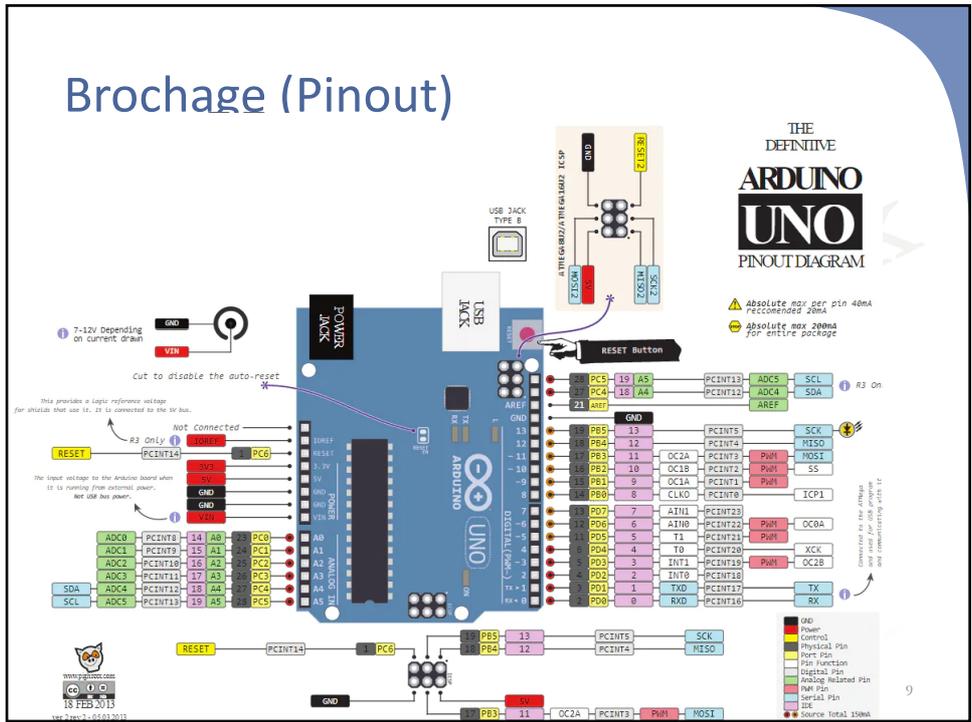
Caractéristiques

- Microcontrôleur ATmega328. cadencé à 16MHz
- 14 entrées/sorties digitales dont 6 permettent d'utiliser la PWM (Pulse Width Modulated)
- 6 entrées/sorties analogiques
- 40mA de courant continu maximum par broche d'I/O (20 mA recommandé et 200mA maximum pour l'ensemble des broches)
- 50mA de courant continu maximum pour la broche de 3.3V
- 32KB de mémoire Flash (0.5KB utilisé par le bootloader)
- 2KB de SRAM
- 1KB d'EEPROM

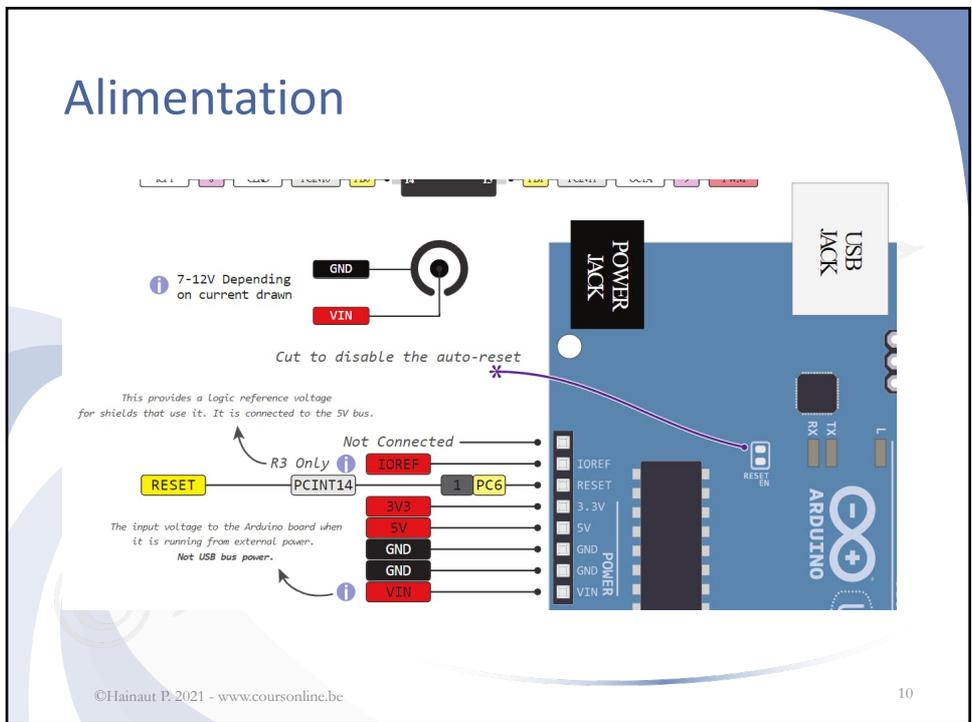
©Hainaut P. 2021 - www.coursonline.be

8

Brochage (Pinout)



Alimentation



Alimentation

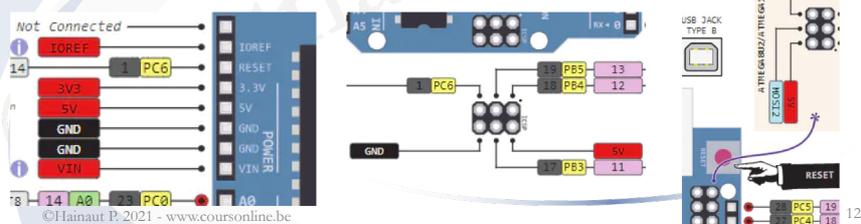
- L'arduino Uno travaille sous 5V et sous 3.3V pour certains cartes d'extension (shield en anglais)
- On peut soit l'alimenter en USB (via le jack USB), ce qui est pratique lorsqu'il connecté à un ordinateur en vue d'une programmation ou d'une vérification de son comportement
- Soit l'alimenter via le power jack avec une tension comprise entre 7 et 12V DC (qui sera convertie en 5V et 3.3V par les circuits de régulation)
- On peut aussi l'alimenter via la broche Vin présente sur la carte (board en anglais)

©Hainaut P. 2021 - www.coursonline.be

11

Alimentation

- La broche Vin est une broche (pin en anglais) d'entrée
- Les broches 5V et 3.3V sont des broches de sortie !
- Il ne faut pas les alimenter !
- Ces broches (couplées avec la broche de masse GND) servent à alimenter les circuits logiques et diverses cartes d'extensions

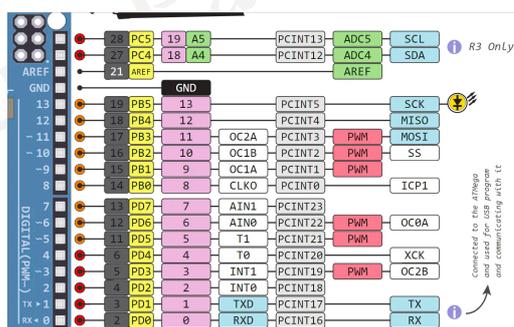


©Hainaut P. 2021 - www.coursonline.be

12

Entrées/sorties digitales

- Ces entrées/sorties travaillent en tout ou rien, avec des signaux logiques 0 ou 1 et des signaux de tension 0V ou +5V
- Sur la broche 13, se trouve la led interne
- Si on met cette borne au niveau 1, la led s'éclaire
- Ca permet d'effectuer des tests sans nécessiter de composant supplémentaire

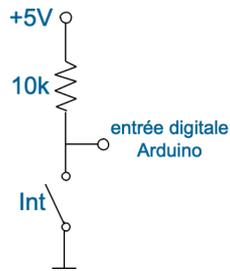


©Hainaut P. 2021 - www.coursonline.be

13

Entrées/sorties digitales

- Si on utilise ces broches en entrée, on doit veiller à ce que l'entrée ne soit pas "flottante" auquel cas l'état logique peut osciller entre les états logiques 0 et 1
- Pour cela, nous utiliserons une résistance de tirage vers le haut (pull-up resistor) qui fixera l'entrée à l'état haut au repos
- L'état par défaut sera donc 1 (+5V)
- L'état actif sera 0 (0V)

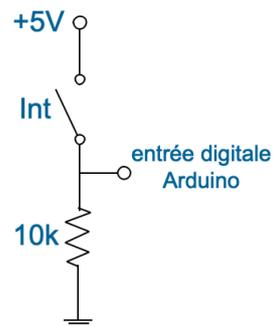


©Hainaut P. 2021 - www.coursonline.be

14

Entrées/sorties digitales

- On peut aussi utiliser une résistance de tirage vers le bas (pull-down resistor) qui fixera l'entrée à l'état bas au repos
- L'état par défaut sera donc 0 (0V)
- L'état actif sera 1 (+5V)



©Hainaut P. 2021 - www.coursonline.be

15

Entrées/sorties digitales: programmation

- En premier lieu, on détermine si la broche est en entrée ou en sortie avec l'instruction:
pinMode(pin, mode);
 - pin: numéro de la broche
 - mode: INPUT ou OUTPUT suivant que la broche est en entrée ou sortie

Exemple: `pinMode(13, OUTPUT);`

©Hainaut P. 2021 - www.coursonline.be

16

Entrées/sorties digitales: programmation

- Ensuite, suivant le mode, on peut lire ou écrire

- Pour lire:
digitalRead(*pin*);
 - *pin*: numéro de la broche

Exemple: `digitalRead(13);`

Le résultat de la lecture sera un état LOW ou HIGH

- On peut affecter le résultat de la lecture à une variable

Exemple: `var=digitalRead(13);`

©Hainaut P. 2021 - www.coursonline.be

17

Entrées/sorties digitales: programmation

- Pour écrire:
digitalWrite(*pin,value*);
 - *pin*: numéro de la broche
 - *value*: HIGH ou LOW

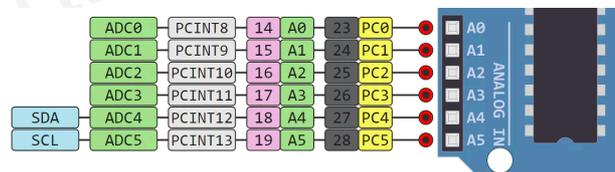
Exemple: `digitalWrite(13,HIGH);`

©Hainaut P. 2021 - www.coursonline.be

18

Entrées analogiques

- Ces entrées travaillent avec des tensions comprises entre 0 et 5V
- Le microcontrôleur travaillant en digital, un convertisseur analogique-digital est associé à ces entrées/sorties
- C'est un convertisseur sur 10 bits, il permet donc de convertir une tension sur 1024 points
- Pour du 5V DC, ça représente une résolution (un pas) de 4,49 mV par point

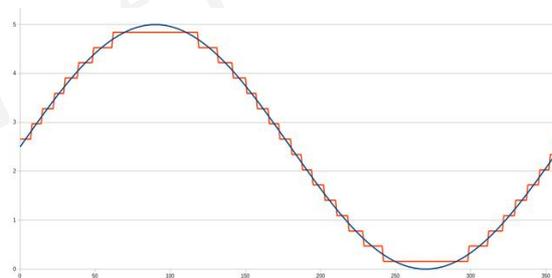


©Hainaut P. 2021 - www.coursonline.be

19

Entrées analogiques

- Le graphique ci-dessous représente la différence entre un signal analogique (il varie de façon continue dans le temps) et un signal numérique (il varie par palier)
- Dans vos programmes, tenez compte de l'imprécision due à la conversion: ne mettez pas comme condition de déclenchement une valeur précise mais un intervalle pour être sûr que le déclenchement s'opère à chaque fois



©Hainaut P. 2021 - www.coursonline.be

20

Entrées analogiques

- Tenez compte aussi qu'il faut plus ou moins 100 μ s pour effectuer la conversion, on peut donc effectuer un maximum de 10.000 mesures par seconde
- Rappel: la tension d'entrée doit être comprise entre 0 et 5V ! sinon vous détruisez la broche
- Une mesure sur une broche non connectée renseigne une tension de quelques centaines de mV car elle agit comme une antenne

Entrées analogiques: programmation

- Pour lire:
analogRead(*pin*);
 - pin: numéro de la broche

Exemple: `analogRead(A0);`

Le résultat de la lecture sera une valeur entre 0 et 1023

- On peut affecter le résultat de la lecture à une variable

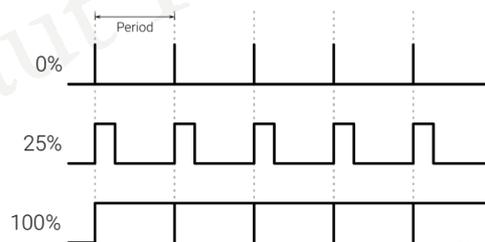
Exemple: `var=analogRead(A0);`

Entrées analogiques: programmation

- Sur l'arduino Uno, il y a un convertisseur analogique/digital (adc) mais pas de convertisseur digital/analogique (dac), donc il n'y a pas de sortie analogique proprement dite
- L'instruction **analogWrite**(*pin,value*) servira donc pour gérer la pwm (dia suivante)

PWM

- On peut aussi utiliser les sorties digitales pour faire varier le signal
- On garde deux niveaux de tension (0 et 5V) mais on va moduler le temps pendant lequel le signal est haut
- A 0%, la tension de sortie est continue et vaut 0V
- A 100%, la tension de sortie est continue et vaut 5V
- Entre les deux, elle est cyclique



PWM

- Avec ce procédé, on peut donc, entre autres, faire varier la vitesse d'un moteur ou la luminosité d'une led (grâce à la persistance rétinienne)
- Les broches permettant la PWM sont les broches 3, 5, 6, 9, 10, 11
- La fréquence (et donc la période) est pré-réglé: 490 Hz sur les broches 3, 9, 10 et 11 et 980 Hz sur les broches 5 et 6

PWM: programmation

- Pour écrire:
analogWrite(pin,value);
 - pin: numéro de la broche
 - value: une valeur entre 0 et 255 où 0 représente un rapport cyclique de 0% et 255 un rapport cyclique de 100%

Si on veut un rapport cyclique de 60%, il faut utiliser une règle de trois pour trouver la valeur à mettre -> $60/100*255$

Exemple: **analogWrite(10,153);**

PWM: programmation

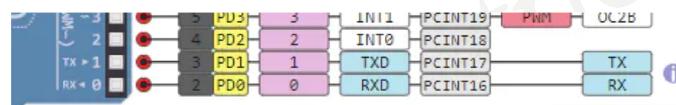
- On peut aussi lire une valeur analogique (par exemple la valeur d'un potentiomètre) et la répercuter en PWM sur une entrée digitale

```
pinMode(10, OUTPUT);  
val = analogRead(A0);  
analogWrite(10, val / 4);
```

La lecture se faisant sur un intervalle de 0 à 1023 et l'écriture sur un intervalle de 0 à 255, on divise par 4 la valeur lue

Communication série

- Les broches 0 et 1 peuvent servir pour communiquer en série avec un autre périphérique



- On peut aussi utiliser le moniteur série de l'environnement arduino pour communiquer avec la carte arduino, ce qui est très pratique lors du débogage de programme

Communication série

- Pour ouvrir le moniteur série:
`Serial.begin(vitesse);`

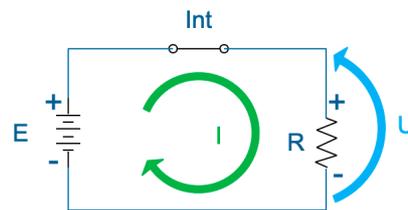
Exemple: `Serial.begin(9600);` vitesse de communication de 9600 bauds

- Pour écrire sur le port série:
`Serial.write();`
- Pour lire sur le port série:
`Serial.read();`
- Pour écrire sur le moniteur série
`Serial.print("I received: ");`

2. NOTIONS D'ELECTRONIQUE ELEMENTAIRE

Loi d'Ohm

- Soit un circuit simple comportant une alimentation continue E (pile, adaptateur secteur, ...) un interrupteur fermé Int , et une résistance R
- La tension aux bornes de la résistance U est égale à la tension d'alimentation $E \rightarrow U=E$
- Un courant circule dans la résistance et il se calcule selon la loi d'Ohm $I=U/R$
- Donc, si $E=9V$, $R=18\Omega$
 $I=9/18=0,5A$
- La tension s'exprime en volts, le courant en ampères et la résistance en ohms

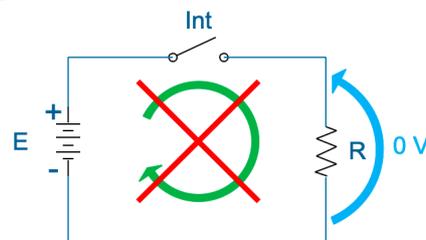


©Hainaut P. 2021 - www.coursonline.be

31

Loi d'Ohm

- Si on ouvre l'interrupteur, le circuit est ouvert et il n'y a plus de courant dans ce circuit
- La tension aux bornes de la résistance est égale à $0V$ ($U=R \times I$ avec $I=0$)
- Toute la tension E se retrouve aux bornes de l'interrupteur

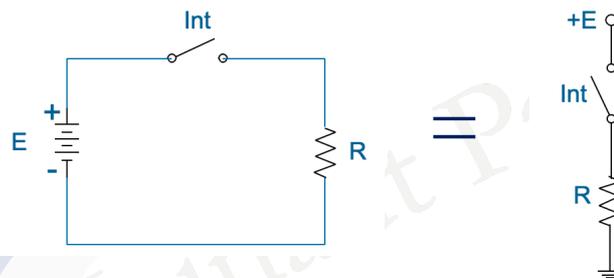


©Hainaut P. 2021 - www.coursonline.be

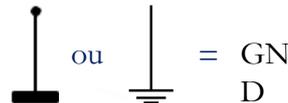
32

Loi d'Ohm: remarque 1

- On peut dessiner différemment le circuit précédent:



- La borne négative de l'alimentation est mise à la masse (GND, 0V)



©Hainaut P. 2021 - www.coursonline.be

33

Loi d'Ohm: remarque 2

- Une tension continue est une tension qui ne varie pas dans le temps
- La plupart des appareils électriques sont alimentés en continu avec une tension d'alimentation bien déterminée (généralement indiquée)



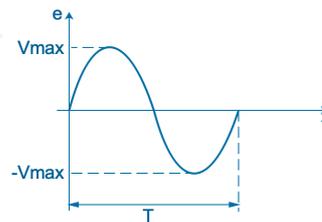
- La valeur de cette tension doit être respectée !
- Ne branchez pas une alimentation de 12V DC sur un appareil prévu pour du 6V DC ! (DC pour Direct Current)

©Hainaut P. 2021 - www.coursonline.be

34

Loi d'Ohm: remarque 2

- Une tension alternative est une tension qui varie dans le temps, souvent en suivant une sinusoïde, on parle alors de signal sinusoïdal
- Exemple: la tension secteur de 220V sinusoïdal, 220V étant la tension moyenne ($V_{\max} = \sqrt{2} \times V_{\text{moy}}$)
- La période T est le temps que met le signal alternatif pour parcourir la sinusoïde
- T est l'inverse de la fréquence: $f = 1/T$
- La tension secteur à une fréquence de 50Hz (en Europe car aux USA, la tension secteur est de 110V 60Hz)

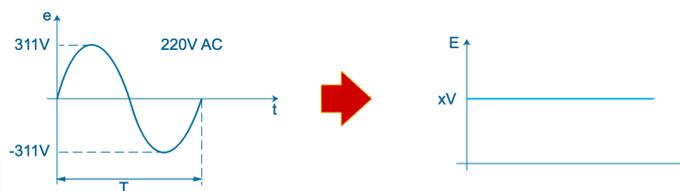


©Hainaut P. 2021 - www.coursonline.be

35

Loi d'Ohm: remarque 2

- On peut abaisser la tension secteur grâce à un transformateur abaisseur de tension
- Si on redresse et si on filtre cette tension, on obtient une tension continue



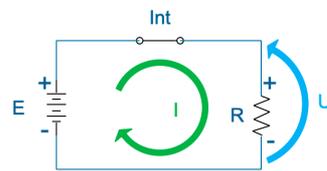
- C'est comme ça que fonctionnent les adaptateurs secteur qui ont une tension d'entrée de 220V AC (AC pour Alternative Current) et qui fournissent une tension de sortie de xV DC (x étant la valeur de la tension de sortie, 5V par exemple)

©Hainaut P. 2021 - www.coursonline.be

36

Loi d'Ohm: remarque 3

- Le courant électrique est comme un débit d'eau dans une tuyauterie, si le robinet est fermé (ce qui correspond à un interrupteur ouvert), l'eau ne circule pas et le débit d'eau est égal à 0
- Si on ouvre le robinet (ce qui correspond à un interrupteur fermé), l'eau coule et il y a un débit d'eau dans la tuyauterie
- Le sens conventionnel du courant électrique va du + vers le -
- Ce courant est en fait un courant d'électrons et va donc du - au +, mais cela ne change rien au principe de fonctionnement, et pour ne pas changer tous les manuels, on a gardé le sens conventionnel



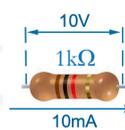
©Hainaut P. 2021 - www.coursonline.be

37

Notion de puissance

- Un courant traversant un composant génère de la chaleur et consomme une certaine puissance
- On la calcule avec la formule: $P=U \times I$ avec P la puissance en watts

- La résistance de gauche est traversée par un courant de $10/1000=0,01A$
Sa puissance doit être d'au moins $10 \times 0,01=0,1W$
- La résistance de droite est traversée par un courant de $10/30=0,33A$
Sa puissance doit être d'au moins $10 \times 0,33=3,3W$

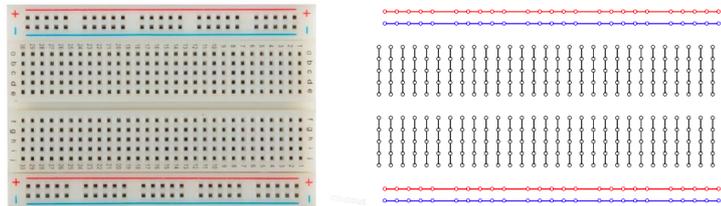


©Hainaut P. 2021 - www.coursonline.be

38

Quelques composants: le breadboard

- Le breadboard (littéralement; la planche à pain) est une plaquette d'expérimentation avec des connexions préétablies, dans laquelle on peut insérer les composants
- Vous voyez sur la figure de droite comment les connexions sont établies au sein du breadboard (sur les breadboards plus petits, les lignes horizontales sont absentes)



©Hainaut P. 2021 - www.coursonline.be

39

Quelques composants: l'interrupteur

- Un interrupteur (int) peut prendre deux états permanents:

- ouvert: le courant ne passe pas 
- fermé: le courant passe 



- Un bouton poussoir (BP) peut prendre deux états, dont un temporaire:

- BP normalement ouverts (NO) 
- BP normalement fermés (NF) 



©Hainaut P. 2021 - www.coursonline.be

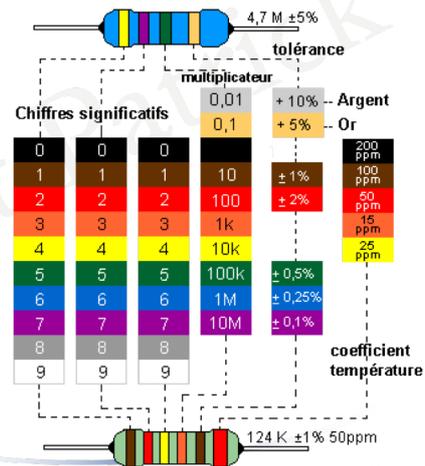
40

Quelques composants: la résistance

- La résistance permet de limiter le courant et/ou la tension



- Un code couleur permet de retrouver sa valeur
- Les résistances utilisées dans le cadre de ce cours, sont des résistances d'1/4 ou 1/2W



©Hainaut P. 2021 - www.coursonline.be

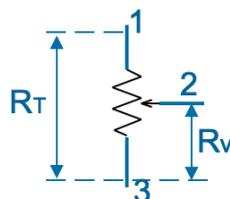
41

Quelques composants: le potentiomètre

- Le potentiomètre est une résistance variable



- Entre les bornes 1 et 3, on a la valeur totale de la résistance
- Entre les bornes 2 et 3, on a une valeur (R_V) qui varie entre 0 et R_T



©Hainaut P. 2021 - www.coursonline.be

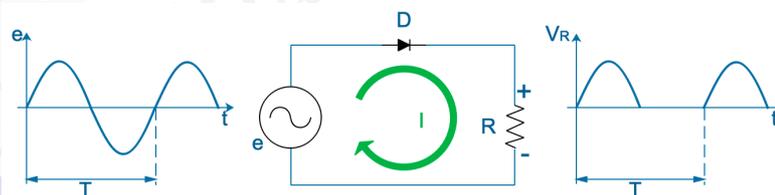
42

Quelques composants: la diode

- La diode est un composant semi-conducteur qui laisse passer le courant dans un sens
- On la retrouve dans les circuits de redressement et de protection



- Exemple d'utilisation:

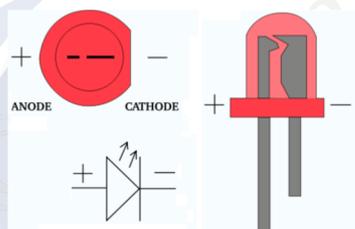


©Hainaut P. 2021 - www.coursonline.be

43

Quelques composants: la LED

- La LED est une diode électroluminescente qui s'éclaire lorsqu'un courant la traverse
- Comme c'est une diode, on doit la brancher dans le bon sens
- Les LEDs standards acceptent un courant nominal de 20 mA
- Le tableau ci-dessous indique la tension aux bornes de la LED



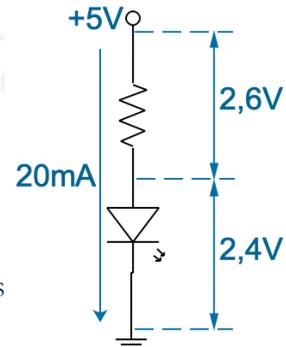
Couleur	Voltage
Rouge	1,6V
Orange	2V
Jaune	2,4V
Verte	2,4V
Bleue	3V
Blanche	3,5V

©Hainaut P. 2021 - www.coursonline.be

44

Quelques composants: la LED

- On doit donc utiliser une résistance en série avec la LED pour prendre l'excès de tension
- On considère une tension d'alimentation de 5V (valeur que l'on trouve sur une broche de sortie de l'arduino) et un courant de 20 mA dans le circuit
- On choisit une diode orange qui prend 2,4V à ses bornes
- La résistance prendra donc 2,6V à ses bornes
- Sa valeur sera de $2,6/0,02=130 \Omega$
- On prendra la valeur du commerce la plus proche (120 ou 150 Ω)



©Hainaut P. 2021 - www.coursonline.be

45

Quelques composants: la ldr

- La ldr (light dependent resistor) est une photorésistance dont la valeur change en fonction de la lumière
- Plus elle est éclairée, plus sa résistance diminue



- On peut s'en servir, par exemple, pour allumer une lampe lorsque le soir tombe

©Hainaut P. 2021 - www.coursonline.be

46

Quelques composants: le micro

- Le micro permet à l'arduino d'interagir avec son environnement sonore



- On peut le câbler sur une entrée analogique ou digitale suivant les besoins
- On trouve des versions "shield" prêtes à l'emploi avec un réglage de sensibilité



©Hainaut P. 2021 - www.coursonline.be

47

Quelques composants: le buzzer

- Le buzzer permet de produire du son
- Deux types:
 - Buzzer actif: il produit un son lorsqu'on l'alimente en continu 
 - Buzzer passif: c'est un haut-parleur élémentaire, il faut lui fournir le signal audio à diffuser 

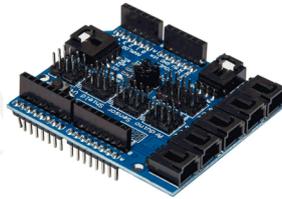
- Attention à la polarité 
- On trouve aussi des versions "shield" 

©Hainaut P. 2021 - www.coursonline.be

48

Les shields

- Les shields sont des cartes d'extension qui se branchent à l'arduino sans soudure, permettant de tester rapidement différents capteurs et d'autres dispositifs comme des drivers moteurs
- On trouve des kits d'expérimentation peu onéreux sur aliexpress



White LED



HC-SR04



Sound Sensor



5V Relay



Push Button



Joystick

©Hainaut P. 2021 - www.coursonline.be

49

3. L'IDE ARDUINO

©Hainaut P. 2021 - www.coursonline.be

50

Environnement de développement

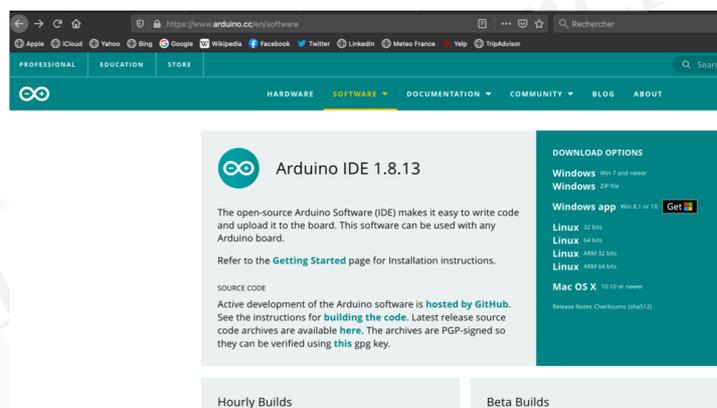
- Pour programmer un arduino, on utilise un environnement de développement (IDE en anglais)
- Une fois le programme terminé, il sera vérifié par l'IDE qui va contrôler la syntaxe, la présence des bibliothèques nécessaires, ...
- On peut ensuite demander à l'IDE de compiler le programme, en ayant indiqué au préalable le modèle de l'arduino utilisé
- Le programme compilé (en langage machine donc) sera finalement écrit dans la mémoire flash de l'arduino
- Attention, il faut garder précieusement le programme source car on ne sait pas le récupérer à partir de l'arduino !

©Hainaut P. 2021 - www.coursonline.be

51

Environnement de développement

- L'IDE peut être téléchargé à l'adresse suivante:
<https://www.arduino.cc/en/software>



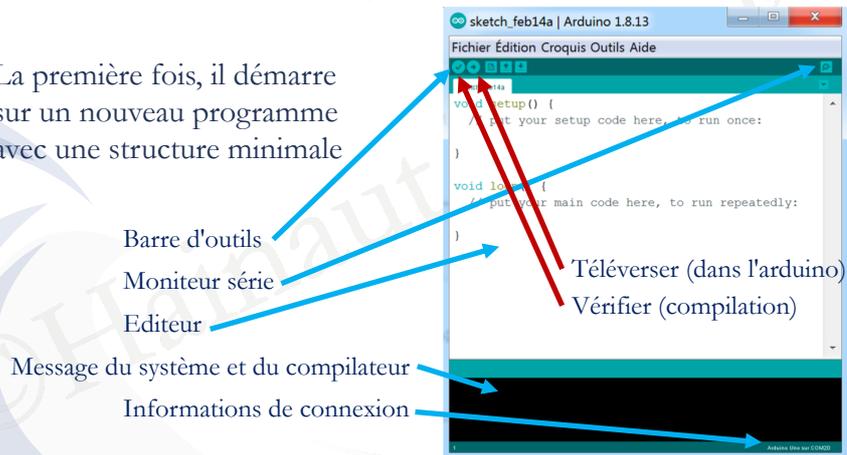
The screenshot shows the Arduino IDE 1.8.13 download page. The page features the Arduino logo and the title 'Arduino IDE 1.8.13'. Below the title, there is a brief description of the IDE and a link to the 'Getting Started' page for installation instructions. A 'SOURCE CODE' section mentions that the code is hosted on GitHub. On the right side, there is a 'DOWNLOAD OPTIONS' section with links for Windows (7 and newer), Windows app (Win 8.1 or 10), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), and Mac OS X (10.10 or newer). At the bottom, there are buttons for 'Hourly Builds' and 'Beta Builds'.

©Hainaut P. 2021 - www.coursonline.be

52

Environnement de développement

- Une fois installé, l'IDE est disponible
- La première fois, il démarre sur un nouveau programme avec une structure minimale

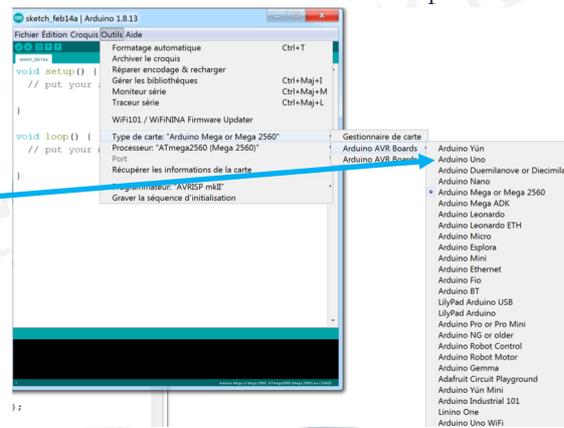


©Hainaut P. 2021 - www.coursonline.be

53

Environnement de développement

- La première chose à faire est de renseigner le modèle d'arduino utilisé, parce que les entrées-sorties et fonctions ne seront pas les mêmes d'un modèle à l'autre
- Dans notre cas, on choisira l'arduino Uno



©Hainaut P. 2021 - www.coursonline.be

54

Environnement de développement

- Au niveau du port série, celui-ci sera émulé à partir de l'USB
- Il faut que le driver USBtoSerial soit bien installé
- Si pas le port reste grisé, une fois que vous avez branché l'arduino au PC
- Si le driver est correctement installé, vous pouvez sélectionner le port

Bases de programmation

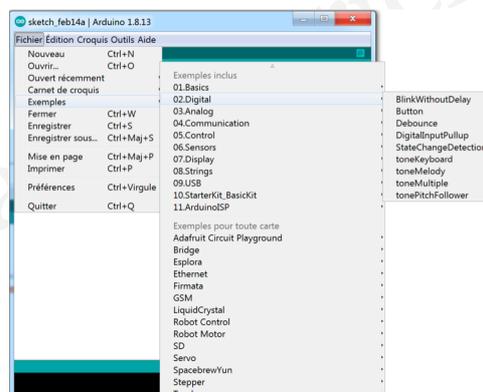
- Un programme comporte au moins deux sections:
 - La section setup qui comprend l'initialisation des entrées/sorties et des variables
 - La section loop qui constitue la boucle principale du programme
L'arduino fonctionnant en autonomie, son programme ne s'arrête jamais et boucle indéfiniment sur lui-même

Bases du langage

- L'arduino se programme en C
- A l'adresse www.arduino.cc/reference/en/ , vous trouvez l'ensemble des instructions acceptées
- Certaines instructions, plus spécifiques, ne fonctionnent que sur certains modèles (mais c'est précisé)
- Nous allons voir quelques les bases de programmation en explorant quelques programmes simples

Exemples

- L'IDE Arduino est livré avec quelques programmes exemples assez simples qui permettent de se faire la main



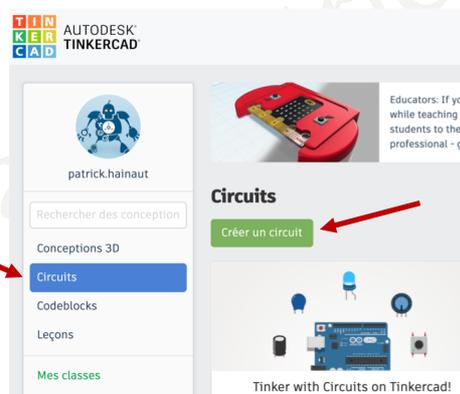
4. ENVIRONNEMENT DE SIMULATION

©Hainaut P. 2021 - www.coursonline.be

59

Tinkercad - circuits

- Tinkercad a évolué et ne propose plus seulement du dessin 3D mais aussi de la simulation microcontrôleurs (actuellement Arduino et micro:bit)

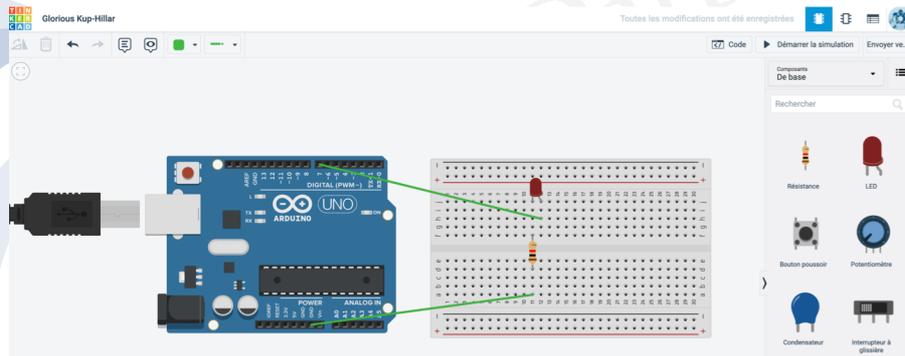


©Hainaut P. 2021 - www.coursonline.be

60

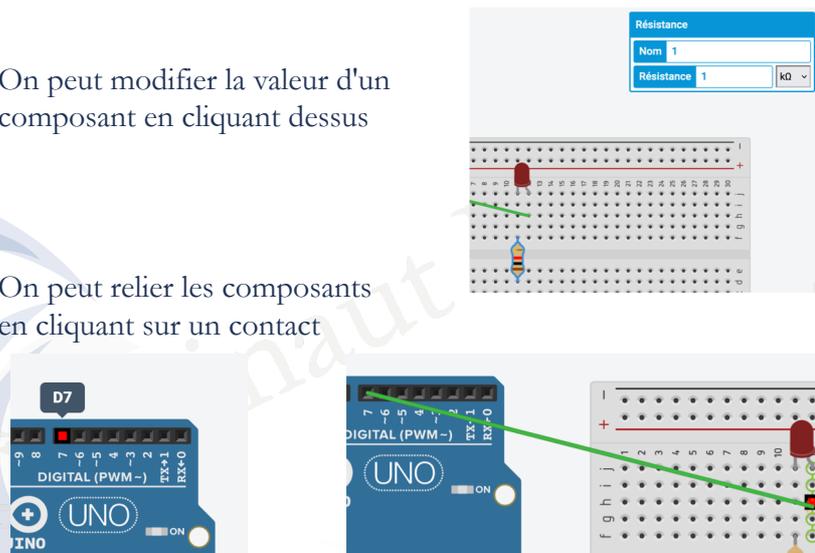
Tinkercad - circuits

- On peut glisser-déposer des composants de la liste de droite sur la feuille d'expérimentation
- La liste des composants disponibles évolue



Tinkercad - circuits

- On peut modifier la valeur d'un composant en cliquant dessus
-
- On peut relier les composants en cliquant sur un contact



Tinkercad - circuits

- Une fois le circuit établi, on peut se pencher sur la programmation, soit par blocs (pour commencer), soit en C
- Le code, produit automatiquement en fonction du schéma construit est bien sur à corriger et adapter

The screenshot shows the Tinkercad interface with the code editor on the left and the block-based programming interface on the right. The code editor contains the following C++ code:

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

The block-based programming interface on the right shows a sequence of blocks: 'définir le voyant LED intégré sur ELEVÉ', 'définir la broche 0 sur ELEVÉ', 'définir la broche 3 sur 0', 'faire pivoter servo sur la broche 0', 'activer le haut-parleur sur la broche 0', 'désactiver le haut-parleur sur la broche 0', 'afficher sur le moniteur série hello world', and 'définir le voyant LED intégré sur FAIBLE'. There are also 'patienter 1 secondes' blocks interspersed.

©Hainaut P. 2021 - www.coursonline.be

63

Tinkercad - circuits

- On peut ensuite lancer la simulation

The screenshot shows the Tinkercad interface with the simulation button highlighted. The button is labeled 'Démarrer la simulation' and is located in the top right corner of the interface. A red arrow points to this button. The circuit diagram below shows an Arduino Uno board connected to a breadboard with a red LED and a resistor. The LED is connected to digital pin 3 and ground, and the resistor is connected to digital pin 0 and ground.

©Hainaut P. 2021 - www.coursonline.be

64

5. EXPÉRIMENTATION

©Hainaut P. 2021 - www.coursonline.be

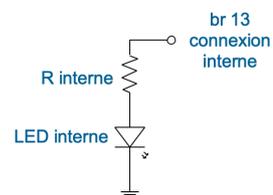
65

Led clignotante

```
void setup() {  
  // initialise la broche digitale en sortie  
  // la pin 13 est connectee a la led interne  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // allume la led  
  delay(1000);           // attend une seconde  
  digitalWrite(13, LOW); // eteint la led  
  delay(1000);          // attend une seconde  
}
```

Matériel:

- Une led sur la br. 13 (digit)
mais elle existe sur le board
et donc pas besoin de
matériel supplémentaire

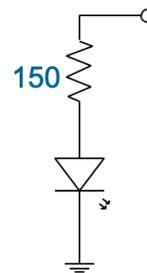


©Hainaut P. 2021 - www.coursonline.be

66

Led clignotante

- Changez les temps d'allumage et d'extinction, prenez des temps différents
- Changez le port de la led et câblez celle-ci en fonction, observez l'effet sur la led interne



©Hainaut P. 2021 - www.coursonline.be

67

Led allumée par poussoir

```
// definition des constantes et numeros de port
const int boutonPin = 2; // le port du bouton poussoir
const int ledPin = 12; // le port de la led
```

```
// definition des variables
int boutonState = 0; // etat du bouton poussoir
```

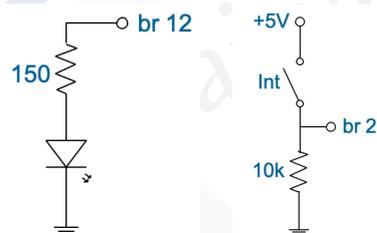
```
void setup() {
  // initialisation du port de la led en mode sortie
  pinMode(ledPin, OUTPUT);
  // initialisation du bouton poussoir en mode entree
  pinMode(boutonPin, INPUT);
}
```

Matériel:

- Une led sur la br. 12 (digit)
- Un BP sur la br. 2 (digit)

```
void loop() {
  // lecture de l'etat du bouton poussoir
  boutonState = digitalRead(boutonPin);

  // regarde si le bouton poussoir a ete pousse
  // si c'est le cas, l'etat est a haut
  if (boutonState == HIGH) {
    // led allumee
    digitalWrite(ledPin, HIGH);
  }
  else {
    // led eteinte
    digitalWrite(ledPin, LOW);
  }
}
```



68

Led allumée par poussoir

- Arrangez-vous pour que la led s'éteigne quand on appuie sur le BP
- Quand on appuie sur le BP, la led doit s'allumer pendant un certain temps (5 sec) puis s'éteindre
- Quand on appuie sur le BP, la led doit s'allumer et s'éteindre quand on réappuie dessus

©Hainaut P. 2021 - www.coursonline.be

69

Led allumée par poussoir

```
const int buttonPin = 2;
const int ledPin = 12;
int buttonState = 0;
int ledState = 0;

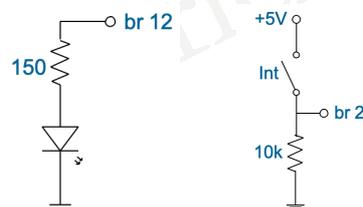
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  ledState = digitalRead(ledPin);

  if (buttonState == HIGH && ledState == LOW) {
    digitalWrite(ledPin, HIGH);
    delay(800);
  }
  else if (buttonState == HIGH && ledState == HIGH) {
    digitalWrite(ledPin, LOW);
    delay(800);
  }
}
```

Matériel:

- Une led sur la br. 12 (digit)
- Un BP sur la br. 2 (digit)



Avec ce programme, la led change d'état à chaque pression du BP

La tempo permet de tenir compte du temps non nul de pression sur le BP (phénomène de rebond)

©Hainaut P. 2021 - www.coursonline.be

70

Led allumée par poussoir

```

const int buttonPin = 2;
const int ledPin = 12;
int ledState = HIGH; // valeur courante de l'état de la LLED
int buttonState; // valeur courante de l'état du poussoir
int lastButtonState = LOW; // valeur précédente de l'état du poussoir

// variables de type long car le temps en millisecondes va prendre une valeur sup à l'int
long lastDebounceTime = 0; // la dernière fois que la broche de sortie a été basculée
long debounceDelay = 50; // délai d'un rebond, à augmenter si ratés

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int reading = digitalRead(buttonPin);

  // si l'état du BP a changé, c'est à cause du rebond ou pas ?
  if (reading != lastButtonState) {
    // réinitialise le compteur de temps avec la valeur courante de temps d'exécution
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    // si la valeur est plus grande que le rebond, alors on peut en tenir compte
    buttonState = reading;
  }

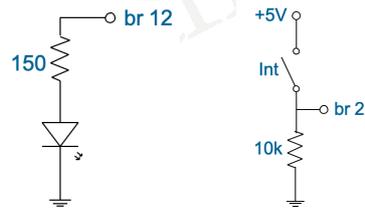
  digitalWrite(ledPin, buttonState);

  // sauve l'état du BP pour la prochaine boucle
  lastButtonState = reading;
}

```

Matériel:

- Une led sur la br. 12 (digit)
- Un BP sur la br. 2 (digit)



Avec ce programme, la led change d'état à chaque pression du BP

Cette version est fonctionnelle à 100%

71

Dimer

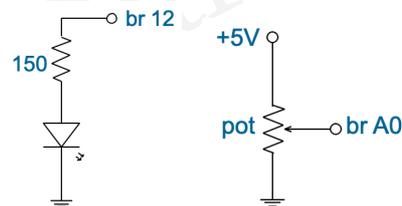
```

// broche de la Led
int ledPin = 12; // broche digitale
// broche du potentiomètre
int potPin = 0; // broche analogique !
// variable stockant la valeur du potentiomètre
int potVal = 0;
void setup()
{
  // mode de la broche digitale
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  // lit la valeur retournée par le pot
  potVal = analogRead(potPin);
  // permet d'ajuster la plage de variation pour correspondre à une plage 0-255
  potVal = potVal/6;
  // écrit une valeur entre 0 (éteint) et 255 (pleine puissance)
  analogWrite(ledPin, potVal);
}

```

Matériel:

- Une led sur la br. 12 (digit)
- Un pot. sur la br. 0 (analog)



72

Dimer

- Changez le facteur de division de postVal par 1, puis par 100 et tirez-en les conclusions

©Hainaut P. 2021 - www.coursonline.be

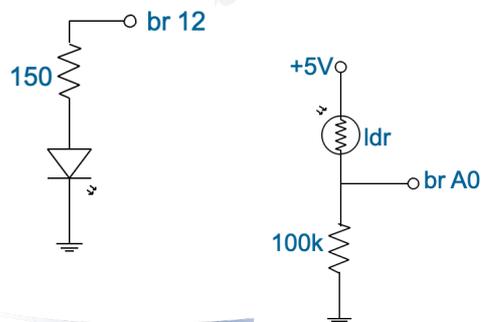
73

Led allumée en fonction de la lumière ambiante

```
// broche de la Led
int ledPin = 12; // broche digitale
// broche de la LDR
int ldrPin = 0; // broche analogique !
// variable stockant la luminosité ambiante lue par la LDR
int lightVal = 0;
void setup()
{
  // mode de la broche digitale
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  // lit la valeur retournée par la LDR
  lightVal = analogRead(ldrPin);
  // allume la LED si la luminosité est faible
  if (lightVal < 700) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    // éteint la LED
    digitalWrite(ledPin, LOW);
  }
}
```

Matériel:

- Une led sur la br. 12 (digit)
- Une ldr sur la br. 0 (analog)



©Hainaut P. 2021 - www.coursonline.be

74

Led allumée en fonction de la lumière ambiante

- Cachez la LDR (résistance variant avec la lumière) et voyez l'effet
- Changez le seuil de lightVal par 300, 500, 800 et 1000
- Tirez-en les conclusions

©Hainaut P. 2021 - www.coursonline.be

75

Fader

```
int brightness = 0; // brillance de la led
int fadeAmount = 5; // palier de progression
```

```
void setup() {
  // la broche 12 sera la broche de sortie
  pinMode(12, OUTPUT);
}
```

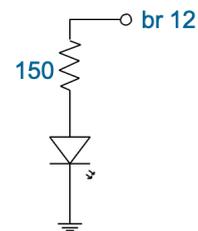
```
void loop() {
  // on écrit sur la broche 12, une valeur analogique, qui sera la brillance de la led
  analogWrite(12, brightness); // la valeur est comprise entre 0 et 255

  // change la valeur de la brillance pour le prochain passage dans la boucle
  brightness = brightness + fadeAmount;

  // si on est arrivé au minimum ou au maximum, on inverse la valeur pour repartir dans l'autre sens
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount;
  }
  // attend 50 millisecondes avant de passer au palier suivant (pour qu'on voie l'effet)
  delay(50);
}
```

Matériel:

- Une led sur la br. 12 (digit)



76

Fader

- Changez le palier avec une valeur de 1, puis 15
- Changez le délai avec une valeur de 20, puis de 100 ms

©Hainaut P. 2021 - www.coursonline.be

77

Fader avec potentiomètre

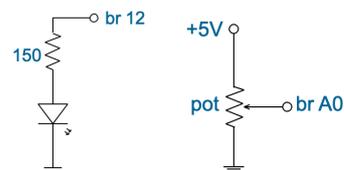
```
int sensorPin = A0; // broche du potentiomètre
int ledPin = 12; // broche de la LED
int sensorValue = 0; // variable contenant la valeur du potentiomètre
int lightVal = 0; // valeur du potentiomètre ajustée à la LED
```

```
void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  // lit la valeur du potentiomètre
  sensorValue = analogRead(sensorPin);
  // ajuste la valeur pour être dans l'intervalle 0-255
  lightVal = sensorValue / 4;
  analogWrite(ledPin, lightVal); // écrit cette valeur sur la broche 12
}
```

Matériel:

- Une led sur la br. 12 (digit)
- Un pot. sur la br. 0 (analog)



78

Fader avec potentiomètre

- Changez le facteur de division de lightVal

©Hainaut P. 2021 - www.coursonline.be

79

Transmission série

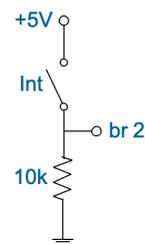
```
void setup() {  
  Serial.begin(9600);  
  pinMode(2, INPUT);  
}  
  
void loop() {  
  int sensorValue = digitalRead(2);  
  Serial.println(sensorValue, DEC);  
}
```

Matériel:

- Un BP sur la br. 2 (digit)

Utilisation:

- Une fois le programme chargé, cliquez sur « Tools », puis « Serial Monitor »



©Hainaut P. 2021 - www.coursonline.be

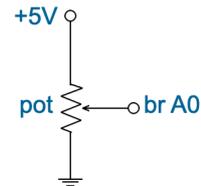
80

Relever la valeur du potentiomètre

```
void setup() {  
  // initialise la communication série:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // envoie la valeur analogique de la broche 0:  
  Serial.println(analogRead(A0));  
  // attend de façon à pouvoir lire le résultat  
  delay(10);  
}
```

Matériel:

- Un pot. sur la br. 0 (analog)



©Hainaut P. 2021 - www.coursonline.be

81

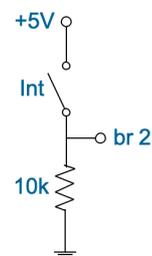
Sonnette

```
// definition des constantes et variables  
const int buttonPin = 2; // le port du bouton poussoir  
const int buzzPin = 10; // le port du buzzer  
int buttonState = 0; // etat du bouton poussoir
```

```
void setup() {  
  pinMode(buzzPin, OUTPUT);  
  pinMode(buttonPin, INPUT);  
}  
void loop() {  
  buttonState = digitalRead(buttonPin);  
  if (buttonState == HIGH) {  
    digitalWrite(buzzPin, HIGH);  
  }  
  else {  
    digitalWrite(buzzPin, LOW);  
  }  
}
```

Matériel:

- Un buzzer sur la br. 10 (digit)
- Un BP sur la br. 2 (digit)

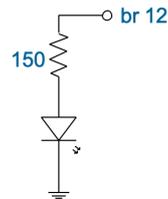


©Hainaut P. 2021 - www.coursonline.be

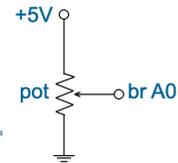
82

Sonnette

- Rajouter la fonction lumineuse à cette sonnette (LED sur broche digitale 12)



- Écrivez un programme qui fasse la même chose que le précédent mais avec le volume du buzzer contrôlé par le potentiomètre connecté en broche analogique 0



©Hainaut P. 2021 - www.coursonline.be

83

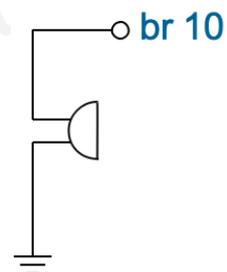
Mélodie

```
Int boucle = 3;
void setup() {
}

void loop() {
  if (boucle != 0) {
    // joue une note sur la broche 10 pendant 200 ms:
    tone(10, 440, 200);
    delay(200);
    // joue une note sur la broche 10 pendant 500 ms:
    tone(10, 494, 500);
    delay(500);
    // joue une note sur la broche 10 pendant 500 ms:
    tone(10, 523, 300);
    delay(300);
    boucle --;
  }
}
```

Matériel:

- Un buzzer sur la br. 10 (digit)



©Hainaut P. 2021 - www.coursonline.be

84

Mélodie

- Téléchargez le programme melody qui se trouve sur www.coursonline.be/pdf/melody.c et testez-le en connectant un buzzer et 2 leds sur les bonnes broches (n'oubliez pas les résistances de limitation de courant pour les leds)

Exercices

- Ecrire un programme qui:
 - Allume ou éteint une led après un « clap » sonore
 - Déclenche une alarme sonore et visuelle si on passe à côté de la platine Arduino
 - Déclenche une alarme sonore et visuelle si on franchit une barrière infrarouge

« Clap » sonore: exemple de solution

```
int ledPin = 12; // broche digitale
int ledState = 0;
int microPin = 0; // broche analogique !
int microVal = 0;
void setup()
{
  // mode de la broche digitale
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  microVal = analogRead(microPin);
  ledState = digitalRead(ledPin);
  if (microVal > 500 && ledState == LOW) {
    digitalWrite(ledPin, HIGH);
    delay (500);
  }
  else if (microVal > 500 && ledState == HIGH) {
    digitalWrite(ledPin, LOW);
    delay (500);
  }
}
```

87

Projet voiture

- L'idée est de construire une voiture autonome pilotée par un arduino
- Le chassis sera dessiné sous TinkerCAD et imprimée en 3D (voir présentation Struct3)
- La voiture aura deux roues motrices à l'arrière et une roue sous forme de bille à l'avant



©Hainaut P. 2021 - www.coursonline.be

88

Projet voiture

- On utilisera un capteur à ultrasons pour détecter les obstacles et un capteur suiveur de ligne pour suivre une trajectoire tracée au sol



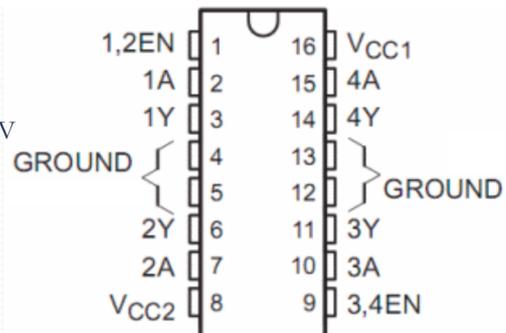
- La voiture pourra être alimentée par un boîtier à piles délivrant 6V en sortie

©Hainaut P. 2021 - www.coursonline.be

89

Projet voiture: contrôle des moteurs

- Cette partie est construite autour d'un circuit intégré, le L293D
 - EN: activation des drivers; actif à +5V
 - 1Y, 2Y: connexion moteur 1
 - 3Y, 4Y: connexion moteur 2
 - 1A, 2A: contrôle moteur 1
 - 3A, 4A: contrôle moteur 2
 - Vcc1: alimentation +5V
 - Vcc2: alimentation: +5 à +24V
 - Gnd: 0V



©Hainaut P. 2021 - www.coursonline.be

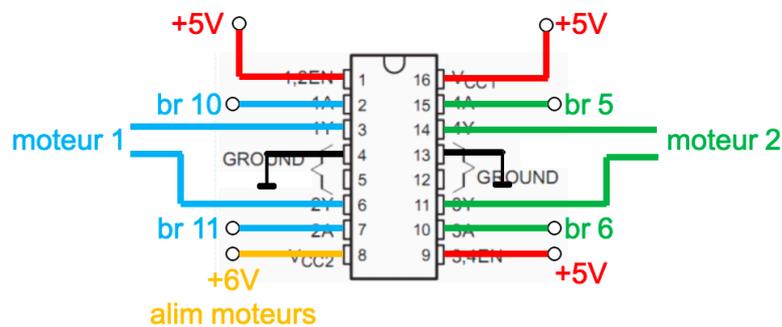
90

Projet voiture: contrôle des moteurs

- Si A1:+5V et A2:0V, le moteur 1 tourne dans le sens 1
Si A1:0V et A2:+5V, le moteur 1 tourne dans le sens 2
Si A1:0V et A2:0V, le moteur 1 s'arrête
- On prendra la sortie 10 pour 1A et la sortie 11 pour 2A
- Même raisonnement pour le moteur 2, avec sortie 5 pour 4A et la sortie 6 pour 3A

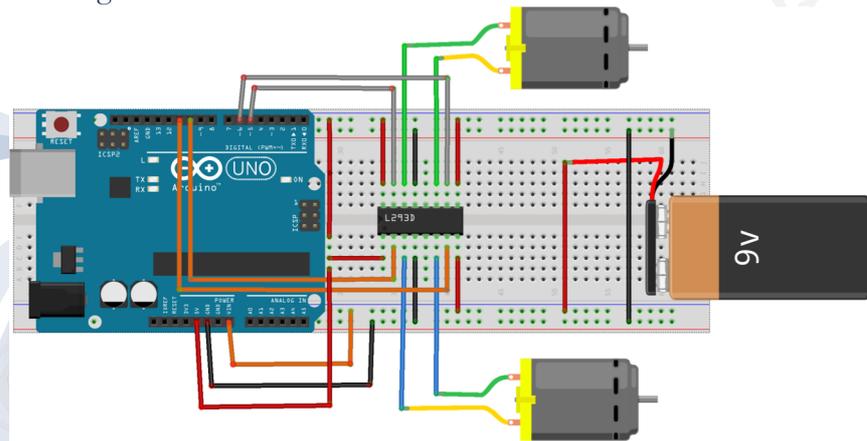
Projet voiture: contrôle des moteurs

- Schéma de principe



Projet voiture: contrôle des moteurs

- Câblage



©Hainaut P. 2021 - www.coursonline.be

fritzing
93

Projet voiture: exemple de code

```
//Motor 1
const int DA = 5; // Pin 15 of L293
const int DB = 6; // Pin 10 of L293

//Motor 2
const int GA= 11; // Pin 7 of L293
const int GB= 10; // Pin 2 of L293

void setup(){
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  //Motor Control - Motor 1: DA, DB & Motor 2: GA, GB
}
```

©Hainaut P. 2021 - www.coursonline.be

94

Projet voiture: exemple de code

```
void ar() { //arrière
  digitalWrite(DA,HIGH);
  digitalWrite(DB,LOW);
  digitalWrite(GA,HIGH);
  digitalWrite(GB,LOW);
}

void av() { //avant
  digitalWrite(DA,LOW);
  digitalWrite(DB,HIGH);
  digitalWrite(GA,LOW);
  digitalWrite(GB,HIGH);
}

void g() { //gauche
  digitalWrite(DA,LOW);
  digitalWrite(DB,HIGH);
  digitalWrite(GA,HIGH);
  digitalWrite(GB,LOW);
}

void d() { //droite
  digitalWrite(DA,HIGH);
  digitalWrite(DB,LOW);
  digitalWrite(GA,LOW);
  digitalWrite(GB,HIGH);
}

void stop() { //stop
  digitalWrite(DA,LOW);
  digitalWrite(DB,LOW);
  digitalWrite(GA,LOW);
  digitalWrite(GB,LOW);
}
```

©Hainaut P. 2021 - www.coursonline.be

95

Projet voiture: exemple de code

```
void loop() {
  av();
  delay(1000);
  ar();
  delay(1000);
  g();
  delay(1000);
  d();
  delay(1000);
}
```

- Ce code permet de gérer uniquement les moteurs
- Ce code doit être modifié dans le cadre de la voiture autonome et la gestion des capteurs est à rajouter

©Hainaut P. 2021 - www.coursonline.be

96

Conclusion

- Nous voilà au bout de cette introduction aux microcontrôleurs
- Pas mal de tutos existent sur Internet
- Le matériel est facilement disponible
- Expérimentez !