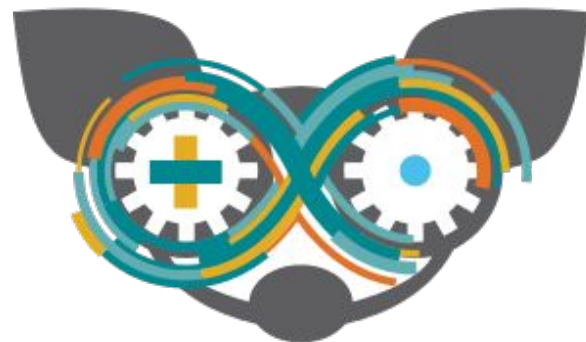


# Arduino - débutant

Les bases, ou comment débuter en douceur...

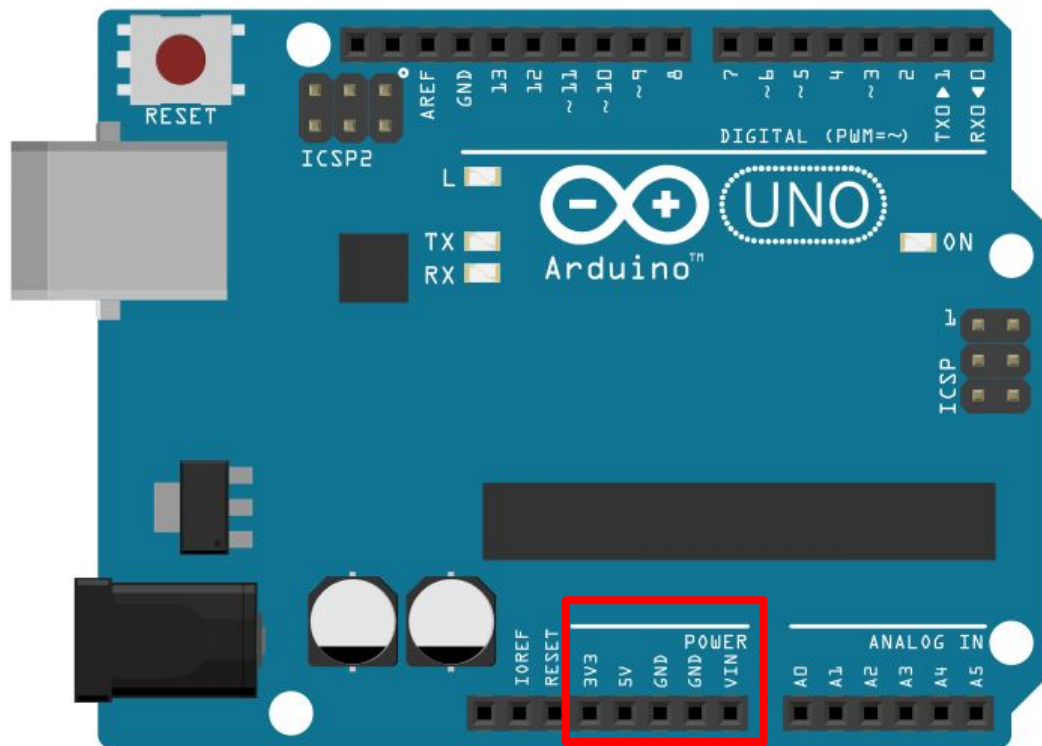


Version du 16 février 2019 - [www.makilab.org](http://www.makilab.org) - CC-BY-SA

# Au menu dans votre kit:

- une carte Arduino (un clone)
- des diodes électroluminescentes (LEDs), rouges, vertes, jaunes
- des résistances (220, 1K, 10K, 100K)
- un “breadboard” & des fils de connection
- des boutons poussoirs
- une photo résistance
- un capteur de température TMP36
- un détecteur de mouvement infrarouge (PIR)
- un buzzer

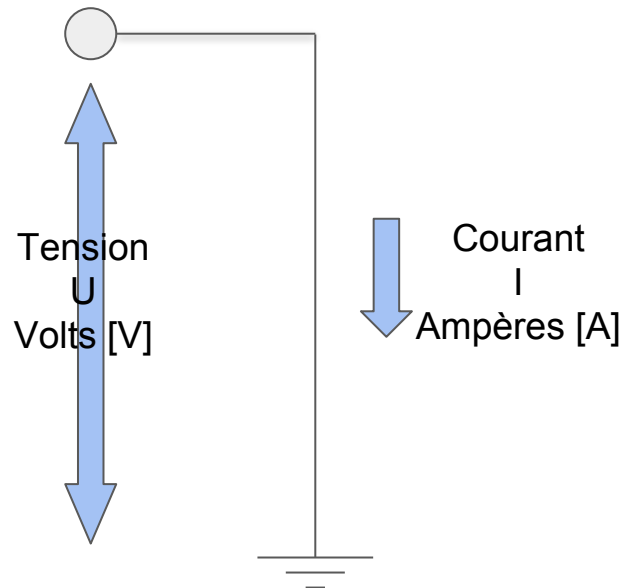
# La carte



# Power on!

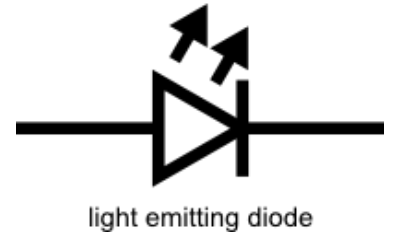
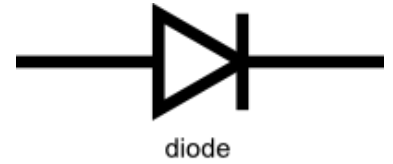
Quelques notions d'électricité:

- C'est une forme d'énergie (comme la chaleur, la lumière, ...)
- Elle se déplace dans des conducteurs (comme les fils)
- On peut la convertir de et vers d'autres formes d'énergie
- AC/DC



# LED the fun begin

Allumons une LED



Anode +    - Cathode

# LED the fun begin

Trop de courant tue une LED,  
en règle générale

max **20mA**

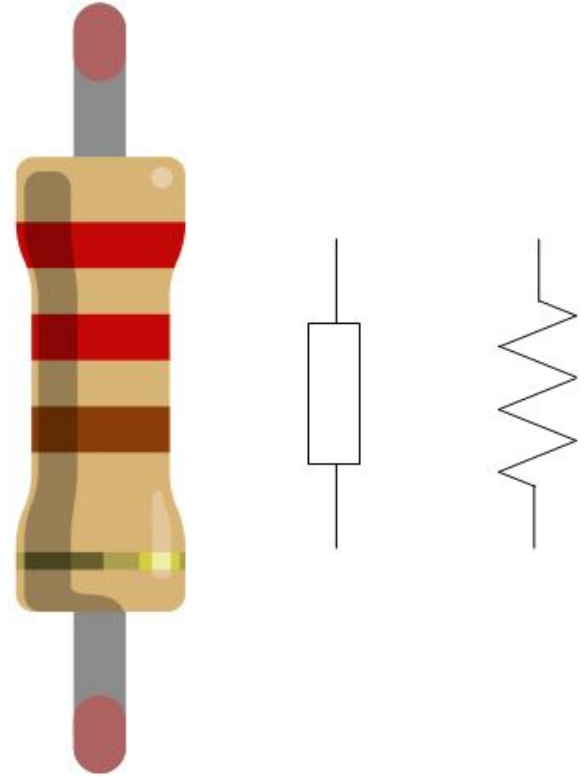
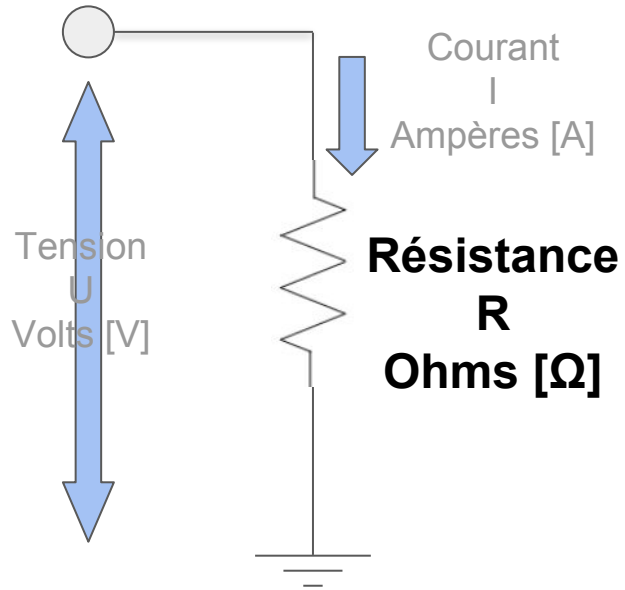
et l'Arduino (Uno) peut supporter jusqu'à

**40mA** par pin I/O

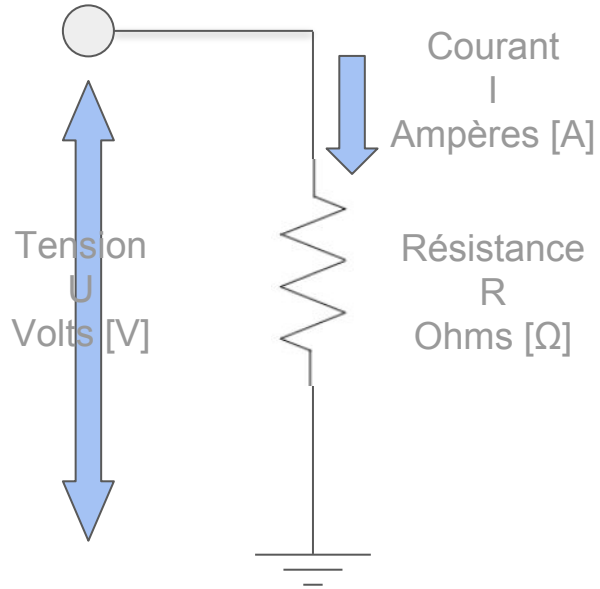
**200mA** par pin VCC ou GND



# Ohm my god



# Ohm my god



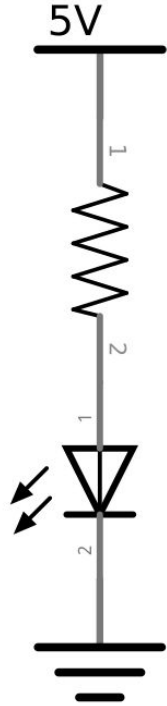
$$U = R * I$$

$$I = U / R$$

$$R = U / I$$



# En pratique



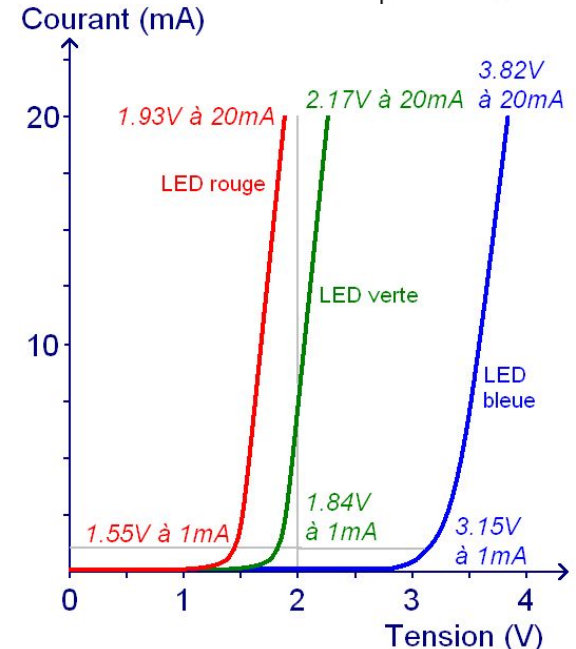
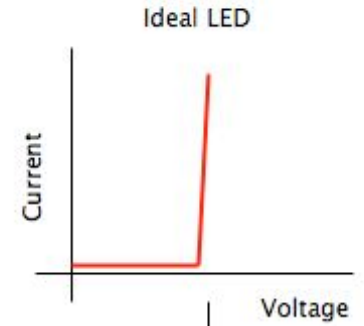
Quel est R pour que I soit  $\leq 20\text{mA}$

# Retour sur la LED

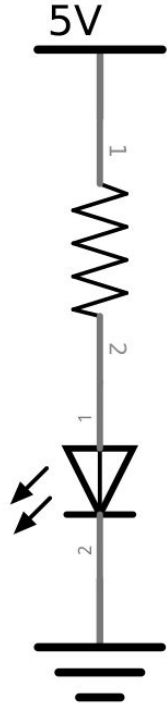
En 1<sup>ère</sup> approximation:

$$R_{LED} \approx 0 \Omega \text{ et } U_{LED} \approx \text{entre } 1.8 \text{ V et } 3.2 \text{ V}$$

DIALIGHT P/N	EMITTED COLOR	MATERIAL	LENS COLOR	LUMINOUS INTENSITY (mcd)			DOMINANT WAVELENGTH (nm)			FORWARD VOLTAGE (V)			VIEWING ANGLE ° DEGREES
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
598-8010-107F	RED	AlInGaP	Water Clear	30	40	80	630	635	642	1.7	2.2	2.4	140
598-8020-107F	RED-ORANGE	AlInGaP	Water Clear	120	150	200	620	625	630	1.7	2	2.4	140
598-8030-107F	ORANGE	AlInGaP	Water Clear	70	-	150	600	-	610	1.7	2	2.4	140
598-8040-107F	YELLOW	AlInGaP	Water Clear	100	130	160	590	-	595	1.7	2	2.2	140
598-8050-107F	YELLOW	AlInGaP	Water Clear	100	130	160	583	-	590	1.7	2	2.4	140
598-8060-107F	YELLOW-GREEN	AlInGaP	Water Clear	20	40	60	570	-	575	1.8	2	2.4	140
598-8070-107F	GREEN	GaP	Water Clear	10	20	40	562	-	570	1.8	2	2.4	140
598-8081-107F	GREEN	InGaN	Water Clear	220	300	400	520	523	525	3	3.2	3.5	140
598-8091-107F	BLUE	InGaN	Water Clear	90	140	160	470	473	475	2.8	3.2	3.5	140



# En pratique

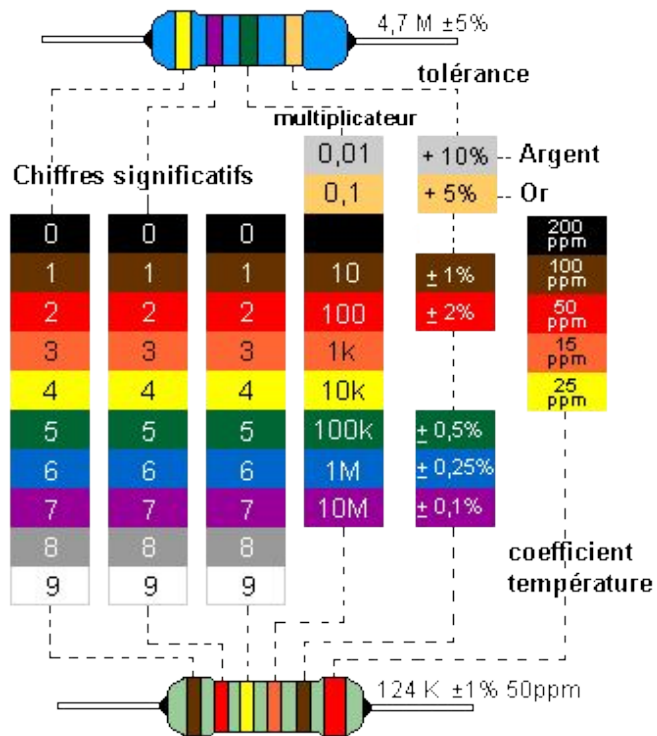


On peut la calculer précisément ...

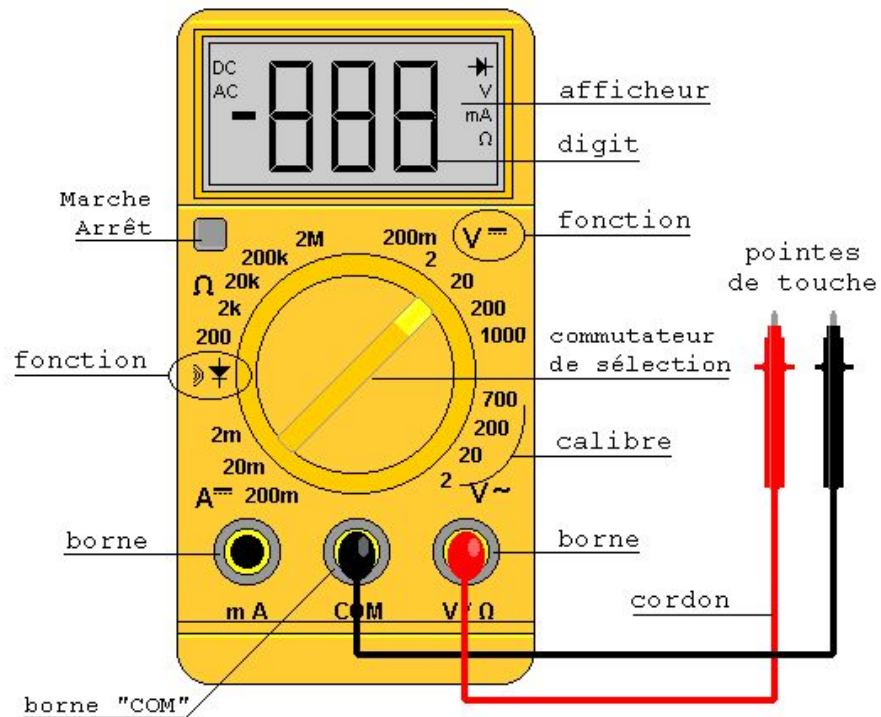
ou retenir que typiquement:

**220**  $\Omega$

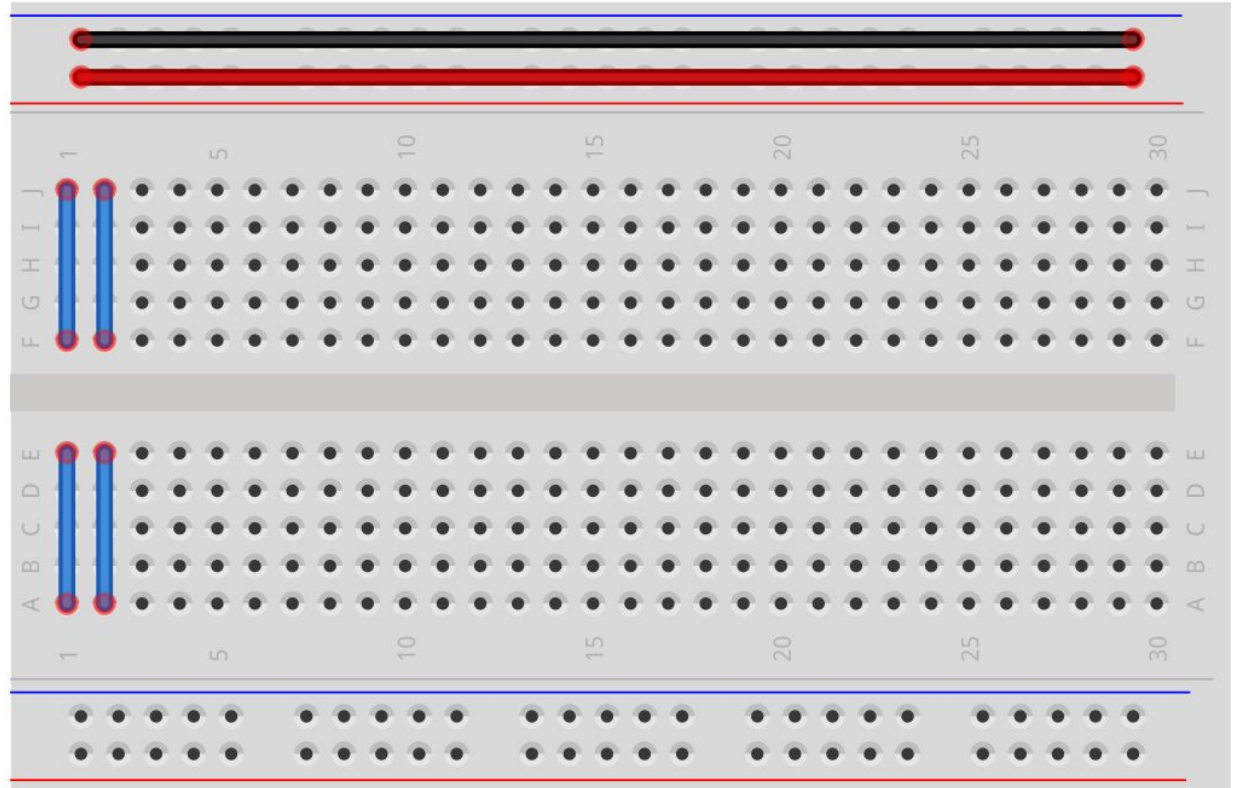
# Ma résistance a quelle valeur?



Ou...



# “Breadboard”

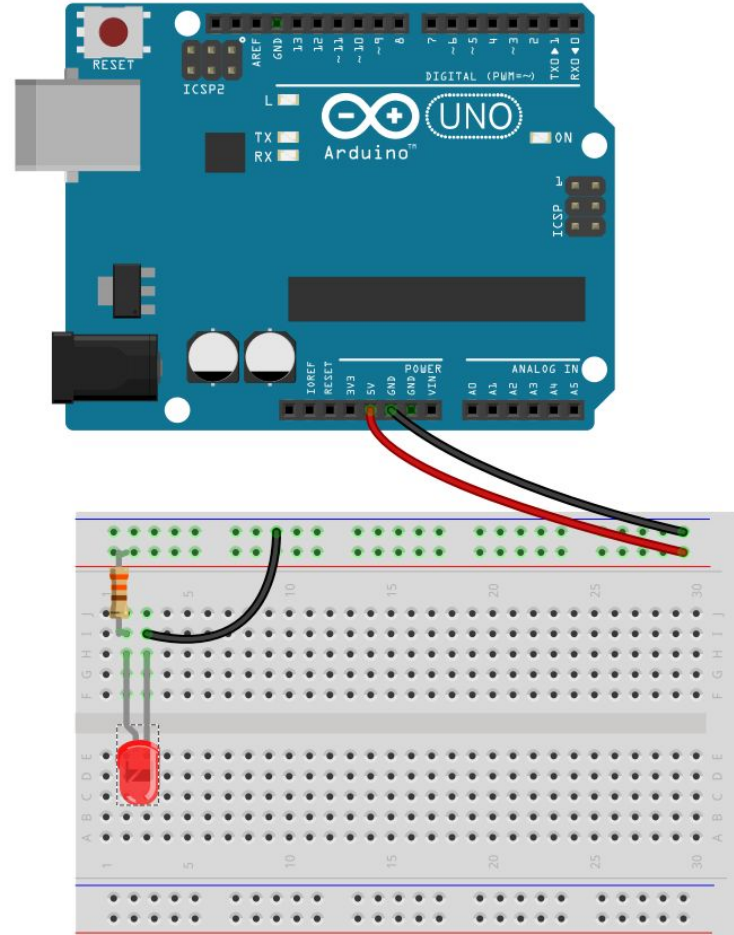


# Circuit LED

- Un circuit doit être “fermé” pour fonctionner.
- Toute l'énergie d'un circuit est utilisée/convertie, donc la tension chute toujours de la tension d'entrée à zéro.
- Le courant électrique cherche toujours le circuit de moindre résistance...



Court-circuit



# Sécurité ?

Faible courants et tensions

Pas de composants capacitifs utilisés aujourd'hui

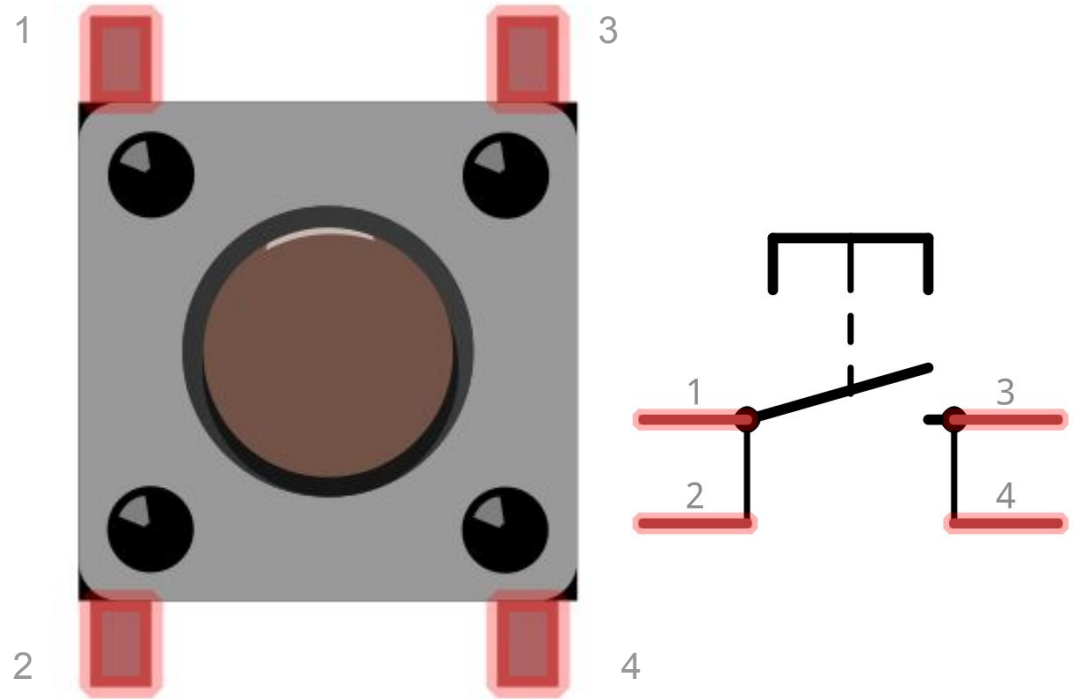


**Pas de risque pour vous...** mais un peu pour le matériel



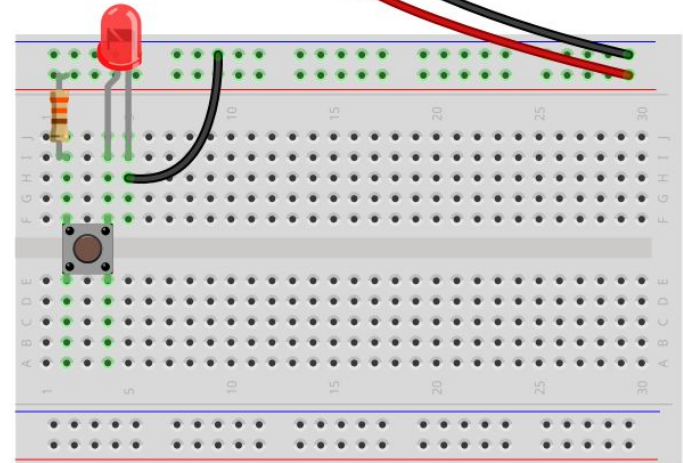
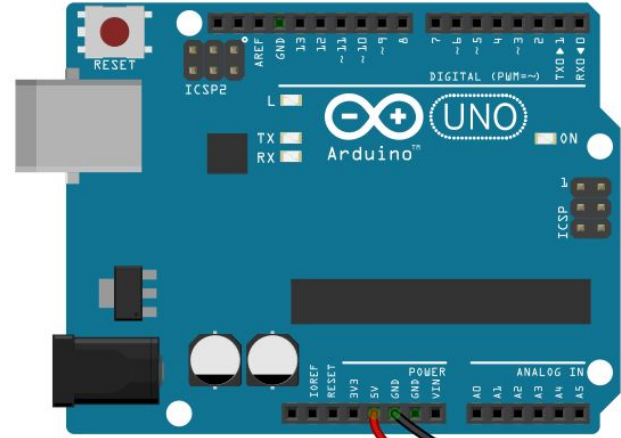
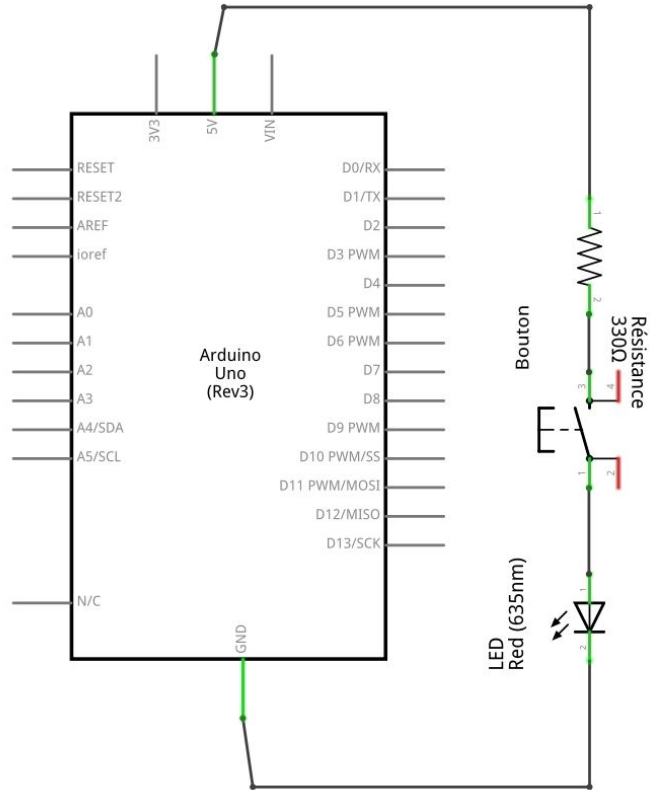
Conseillé de travailler hors tension  
& de vérifier le montage avant mise sous tension

# On / Off

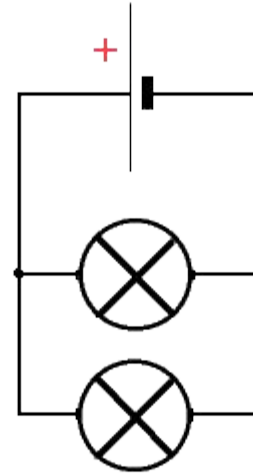
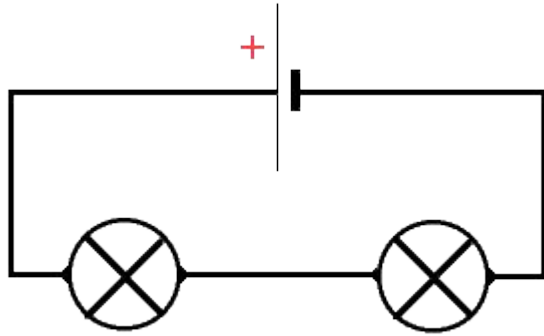




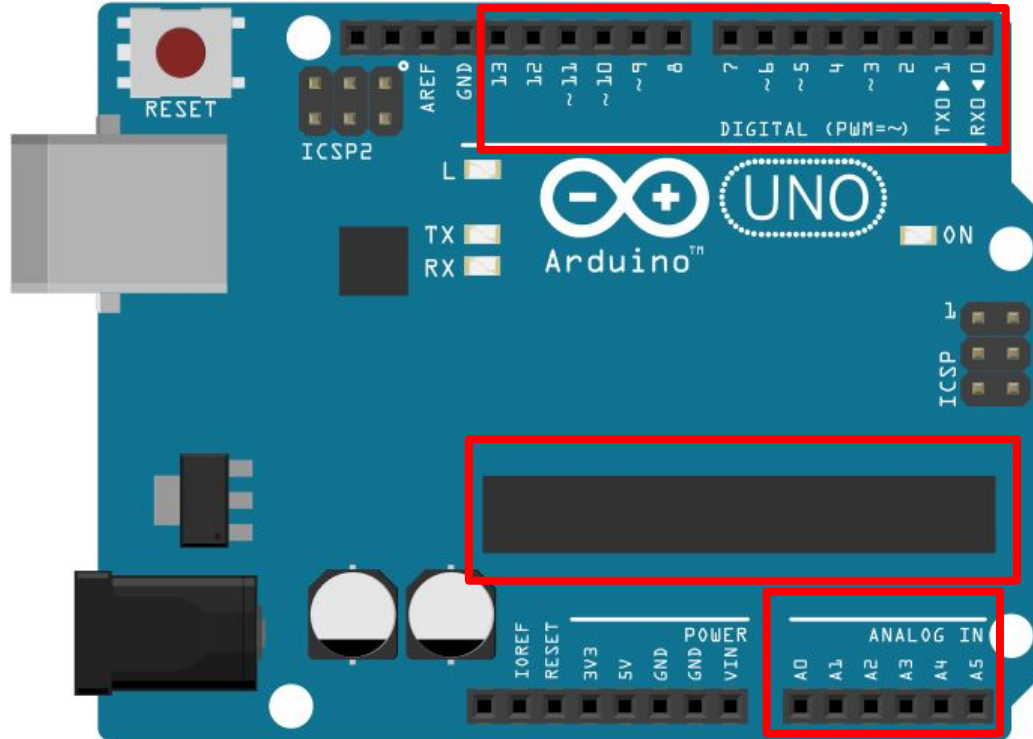
# Circuit On/Off



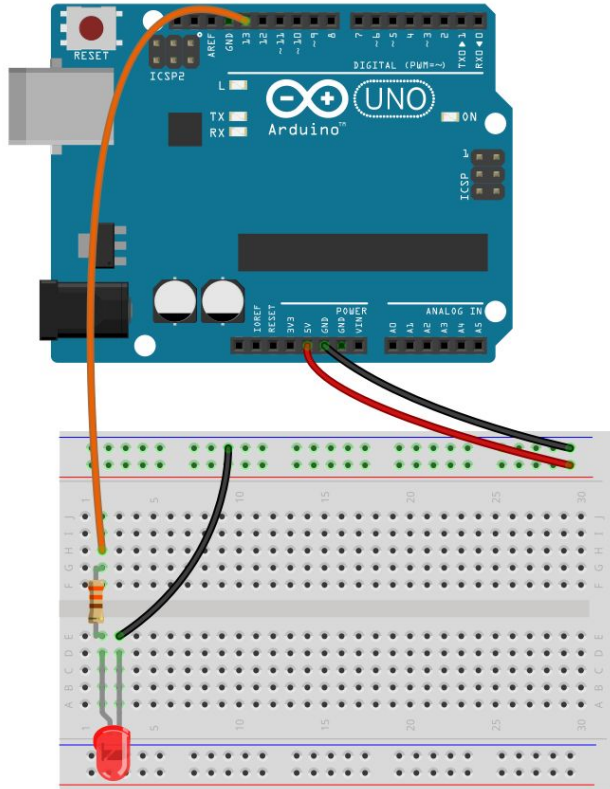
# Série vs Parallèle



# Un microcontrôleur? Des pin I/O?

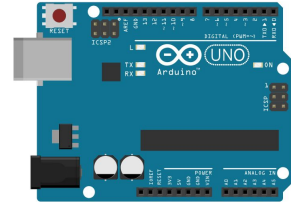
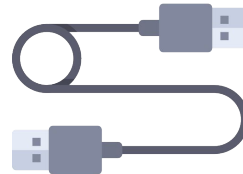
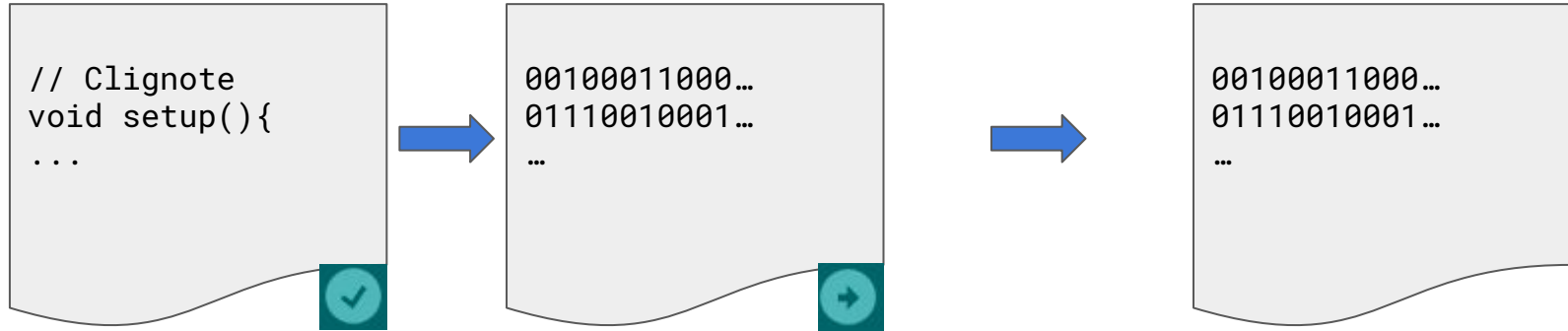


# Allumons Programmons l'allumage d'une LED

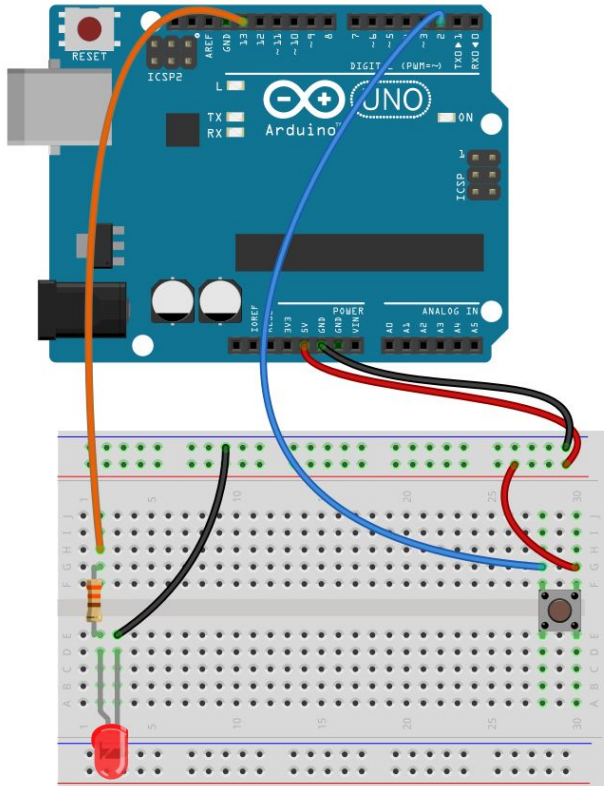


```
Clignote | Arduino 1.8.8
File Edit Sketch Tools Help
Clignote
1 /* La fonction setup s'exécute une seule fois lorsque l'on alimente la
2  * carte, ou que l'on appuie sur le bouton "reset" */
3 void setup() {
4   // initialise la pin digitale 13 comme sortie
5   pinMode(13, OUTPUT);
6 }
7
8 /* La fonction "loop" s'exécute en boucle à l'infini... tout du moins tant
9  * que la carte est alimentée */
10 void loop() {
11  digitalWrite(13, HIGH); // allume la LED (HIGH soit 5V)
12  delay(2000);           // attend 2000 ms (2 secondes)
13  digitalWrite(13, LOW); // éteint la LED (LOW soit 0V)
14  delay(500);           // attend 500 ms (1/2 seconde)
15 }
Done Saving.
Sketch uses 950 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local va
2
Arduino/Genuino Uno on /dev/ttyUSB0
```

# Compilons



# Ajoutons un bouton



```
Echier Édition Croquis Outils Aide
ClignoteBouton | Arduino 1.8.8
Vérifier

ClignoteBouton
1 /* La fonction setup s'exécute une seule fois lorsque l'on alimente la
2  * carte, ou que l'on appuie sur le bouton "reset" */
3
4 int etatBouton = 0;
5
6 void setup() {
7   // initialise la pin digitale 13 comme sortie
8   pinMode(13, OUTPUT);
9   // initialise la pin digitale 2 comme entrée
10  pinMode(2, INPUT);
11 }
12
13 /* La fonction "loop" s'exécute en boucle à l'infini... tout du moins tant
14  * que la carte est alimentée */
15 void loop() {
16   etatBouton = digitalRead(2); // lit l'état du bouton
17   digitalWrite(13, etatBouton); // allume la LED (HIGH soit 5V)
18 }

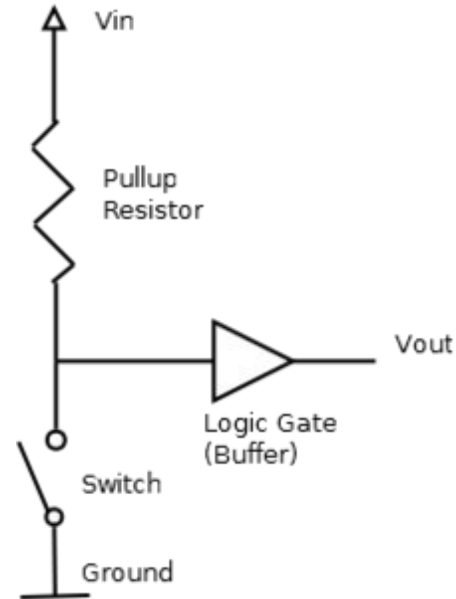
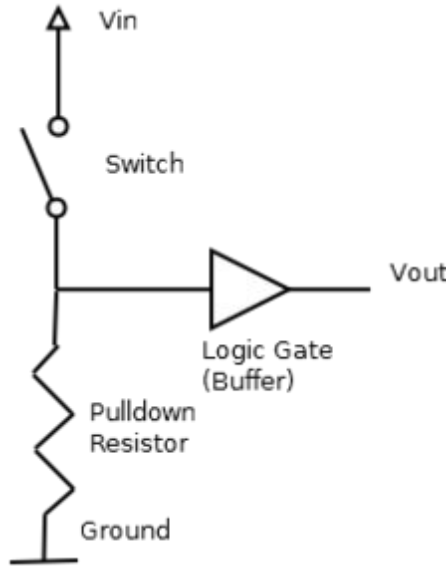
Compilation terminée.
Le croquis utilise 890 octets (2%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2048 octets libres.

13 Arduino/Genuino Uno sur /dev/ttyUSB0
```

# Résistance de tirage



Une pin d'entrée sur laquelle rien n'est connecté n'est pas forcément à LOW (0 V)



# Console série:

```
void setup (){  
    Serial.begin(9600);    // "ouvre" la console série  
}  
  
...  
  
void loop () {  
    Serial.print("Test console série");  
}  
}
```



# Entrée digitale - capteur de mouvement PIR

## Passive Infrared Sensor

Wikipedia [ [https://en.wikipedia.org/wiki/Passive\\_infrared\\_sensor](https://en.wikipedia.org/wiki/Passive_infrared_sensor) ]



Passif -> Il faut l'alimenter en 5V.

Sortie à HIGH si mouvement détecté (et reste HIGH pendant ~8 à 10s)

Sortie à LOW si pas de mouvement

# Entrée digitale - capteur de mouvement PIR



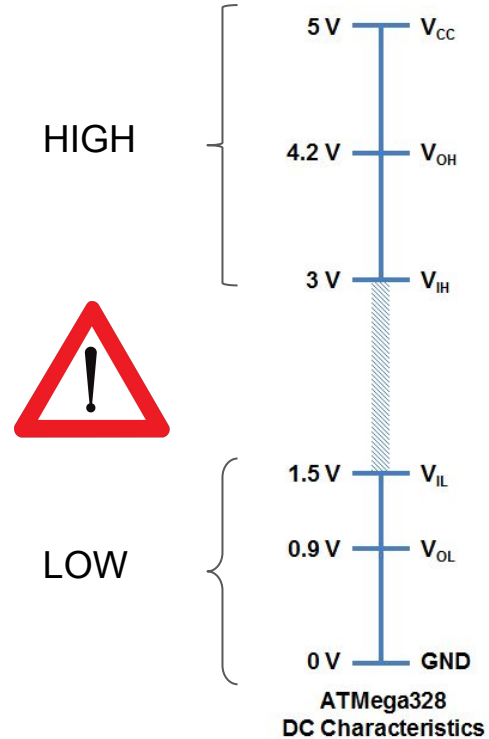
## Passive Infrared Sensor

Wikipedia [ [https://en.wikipedia.org/wiki/Passive\\_infrared\\_sensor](https://en.wikipedia.org/wiki/Passive_infrared_sensor) ]

```
...  
  
void loop () {  
  if (digitalRead(pinPIR) == HIGH) {  
    // Mouvement détecté  
    Serial.print("Mouvement détecté");  
  }  
  else {  
    Serial.print("Pas de mouvement depuis plus 8s");  
  }  
}
```

# Encore un petit mot sur les entrées digitales

Attention à la zone grise entre les niveaux HIGH et LOW



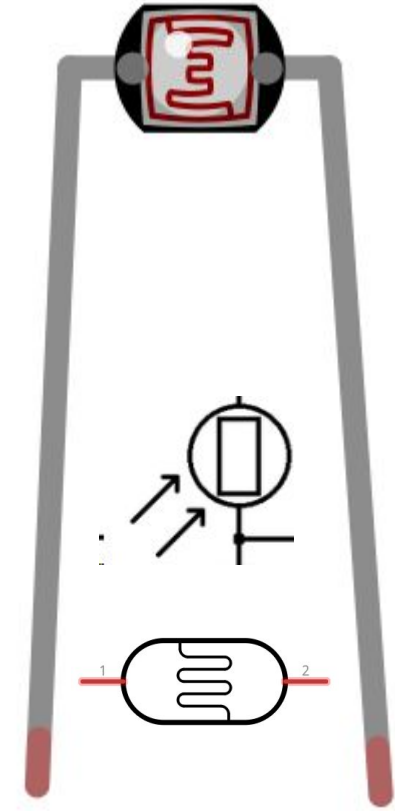
# Entrée analogique - photorésistance

Activité ou lieu concerné	Éclairage moyen
Sensibilité d'une caméra	0,001 lux
Nuit de pleine lune	0,5 lux
Rue de nuit bien éclairée	20 à 70 lux
Local de vie	100 à 200 lux
Appartement bien éclairé	200 à 400 lux
Local de travail	200 à 3 000 lux
Stade de nuit (suivant les différentes catégories : E1, E2, E3, E4, E5)	150 à 1 500 lux
Extérieur par ciel couvert	500 à 25 000 lux
Extérieur en plein soleil	50 000 à 100 000 lux

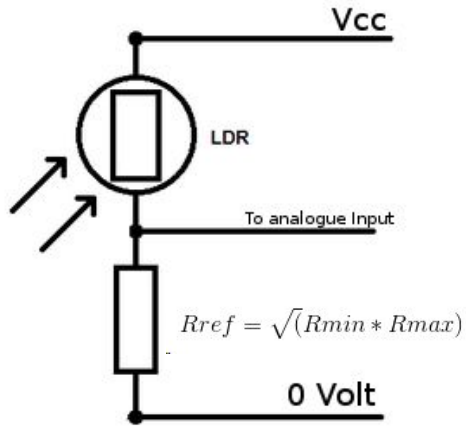
~ 3 MΩ

~ 500 Ω

Source: Wikipedia [ [https://fr.wikipedia.org/wiki/Lux\\_\(unit%C3%A9\)](https://fr.wikipedia.org/wiki/Lux_(unit%C3%A9)) ]



# Photorésistance - montage



```
...  
  
void loop () {  
    valeurPhotoresistance = analogRead(pinPhotoresistance);  
    Serial.print("Valeur du niveau de lumière: ");  
    Serial.print(valeurPhotoresistance);  
}
```

tension **0 ... 5v**

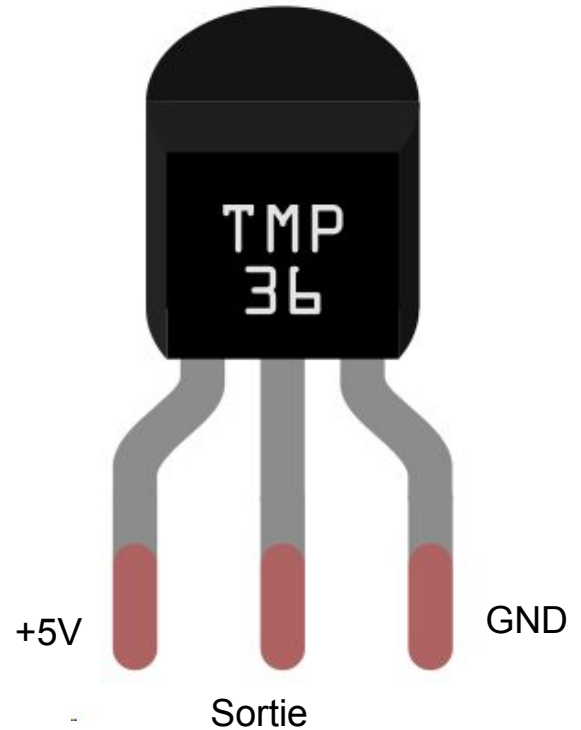
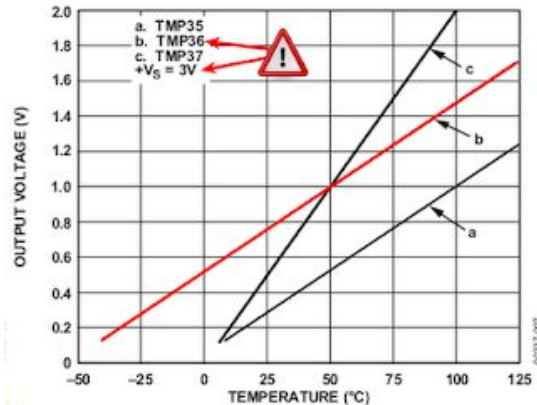


entier **0 ... 1023**

soit une résolution de **4.9mV** (5 volts / 1024) par unité

# Entrée analogique - capteur TMP36

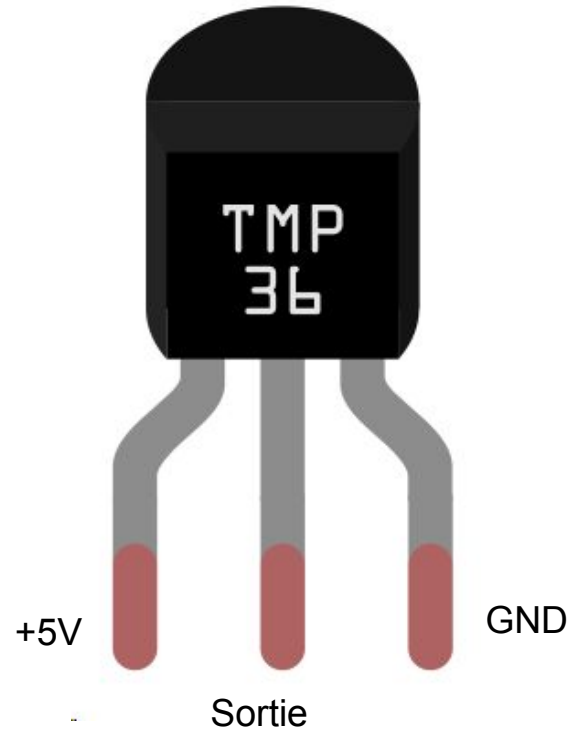
- Tension de sortie: 0.1V (-50°C) to 2.0V (150°C)  
mais la précision diminue après 125°C  
→ 0°C correspond à 0.5V
- Tension d'alimentation: 2.7V a 5.5V
- Courant de charge: 0.05 mA



# Entrée analogique - capteur TMP36

- Tension de sortie: 0.1V (-50°C) to 2.0V (150°C)  
mais la précision diminue après 125°C  
→ 0°C correspond à 0.5V
- Tension d'alimentation: 2.7V a 5.5V
- Courant de charge: 0.05 mA

```
...  
void loop () {  
    ...  
    valeurTMP36 = analogRead(pinTMP36);  
    tensionTMP36 = (valeurTMP36/1024.0) * 5.0;  
    valeurTemperature = (tensionTMP36 - 0.5) * 100;  
    Serial.print("Valeur de température [°C]: ");  
    Serial.print(valeurTemperature);  
    ...  
}
```



# Sortie analogique - buzzer

- Tension d'entrée: 3 à 30V
- Fréquence entre 2kHz et 10kHz

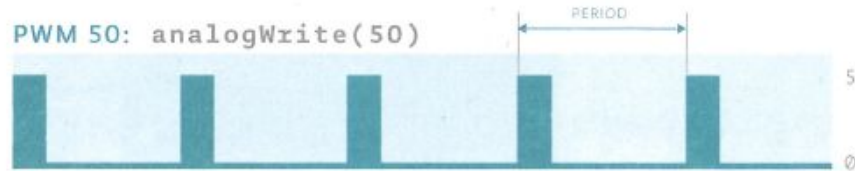


```
...  
void loop () {  
  ...  
  // joue un son de 3kHz pendant 20 millisecondes  
  tone(pinPiezzo, 3000, 20);  
  ...  
}
```

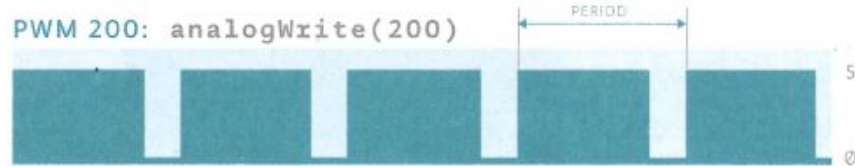


# Tone vs PWM

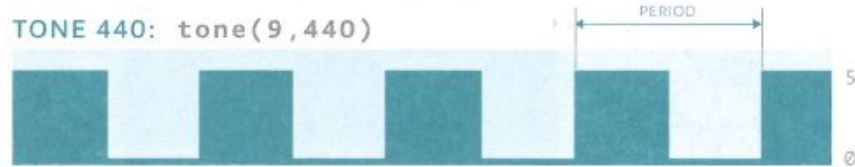
Notice how the signal is low most of the time, but the frequency is the same as PWM 200.



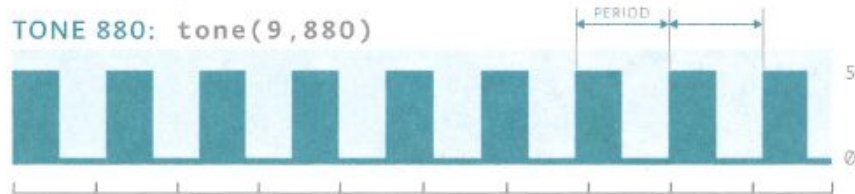
Notice how the voltage is high most of the time, but the frequency is the same as PWM 50.



The duty cycle is 50% (on half the time, off half the time), but the frequency changes.



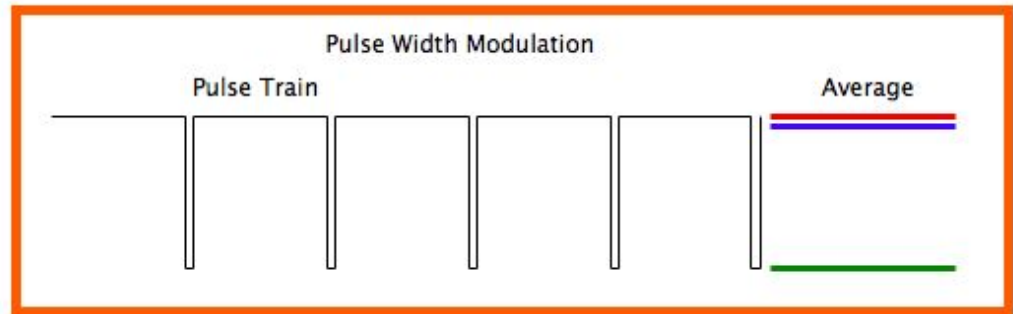
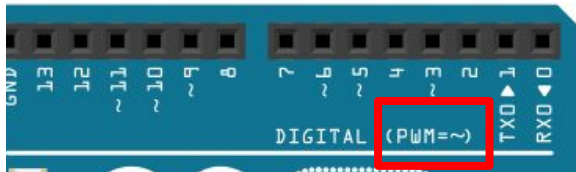
Same duty cycle as Tone 440; but twice the frequency.



# PWM: Pulse Width Modulation

Permet d'approximer un niveau "analogique" par moyenne temporelle

```
..  
  analogWrite(cycleDeTravail);  
  // Cycle de travail entre 0 (toujours LOW) et 255 (toujours HIGH)  
..
```



# That's all folks

Formation faite au Makilab ([www.makilab.org](http://www.makilab.org) - [info@makilab.org](mailto:info@makilab.org) - Quentin Berten)

Diffusé sous licence



(<http://creativecommons.org/licenses/by-sa/4.0/>)