

Atelier Initiation à l'Arduino

github.com/
alexAubin/
atelierArduino



Alexandre Aubin
et Nicolas Chesnais
(Hackstub)

Les Hacketliers
7 & 8 février 2014

Pourquoi cet atelier

« À l'heure où la technologie impacte de plus en plus les aspects sociaux, politiques et culturels de nos vies,

il est crucial que celle-ci reste comprise et contrôlée par les citoyens

et non qu'elle devienne un instrument de pouvoir pour ceux qui la construise. »

Objectifs

- Comprendre les principes de bases de l'électronique et de la programmation
- Réaliser des montages avec Arduino
- Acquérir les connaissances pour démarrer son propre projet

Programme

- Bases de l'électronique (30~40 min)
- Votre tout premier montage Arduino (30~40 min)
- Construction et programmation d'un **robot** (45min~1h)

(1)

Les bases de l'électronique

- Définition
- Tension, intensité
- Les composants
- Arduino dans tout ça
- La programmation

Definition

Electronique

↔

Hydraulique

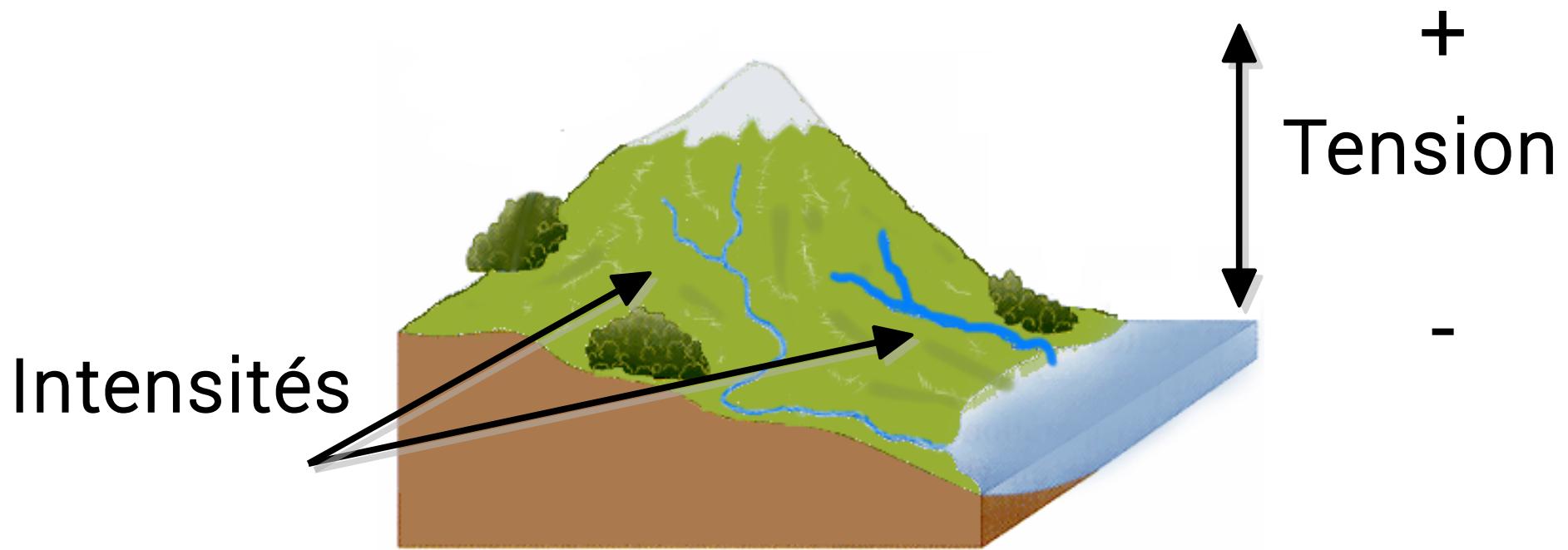
Manipuler
des électrons
avec des fils

Manipuler
de l'eau
avec des tuyaux

Tensions, intensité

Tension ~ Pression/hauteur

Intensité ~ Débit

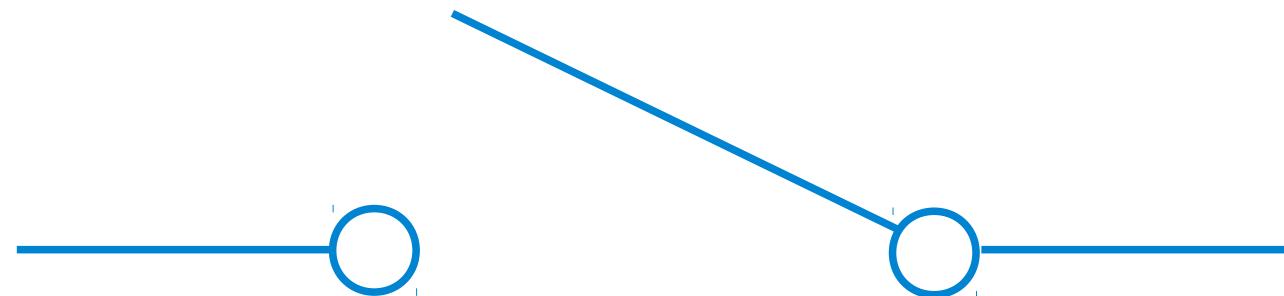


Interrupteur



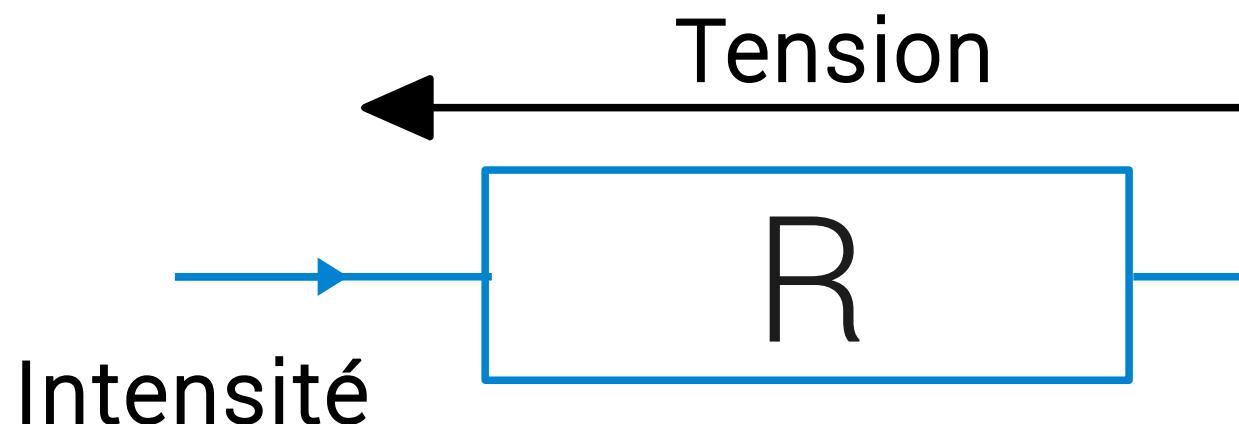
Porte / robinet

Pour donner du contrôle à l'utilisateur

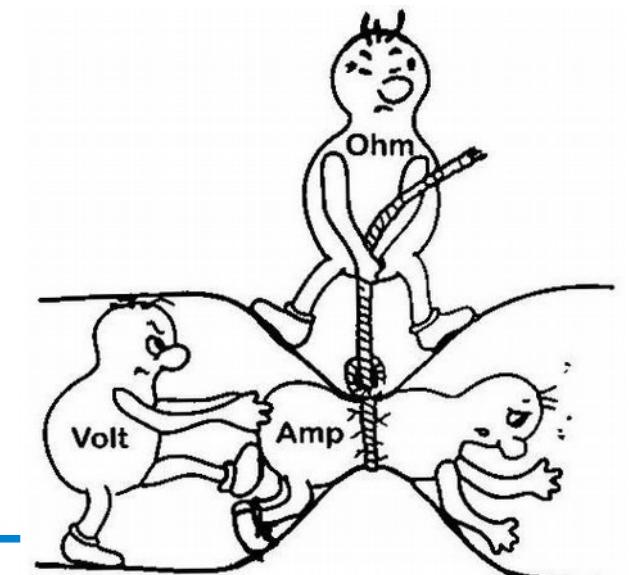


Résistance

Protège les composants d'un débit trop élevés),



$$\text{Tension} = R \times \text{Intensité}$$



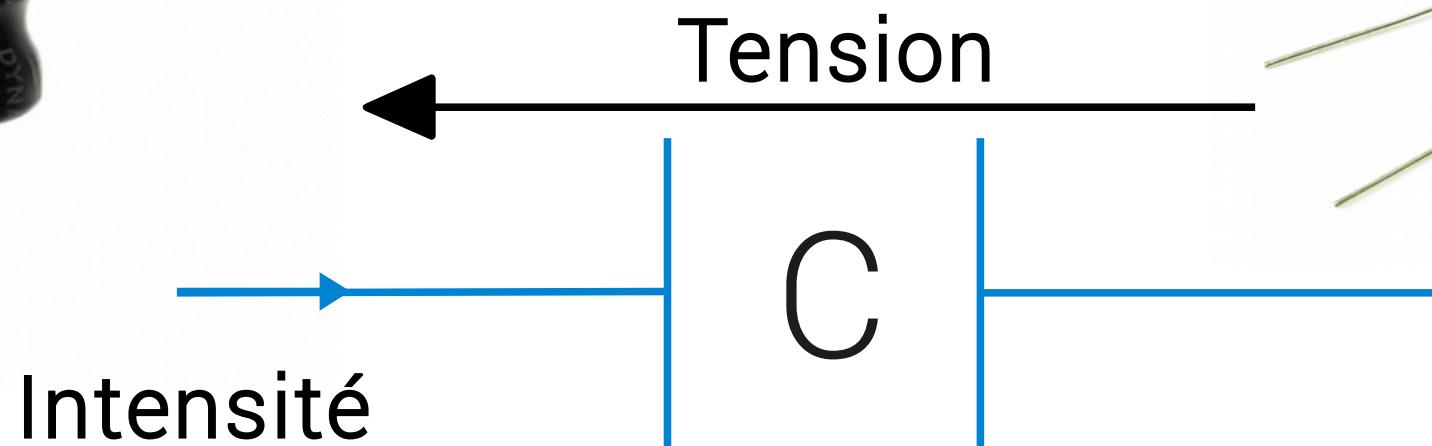
Certaines résistances dépendent de la température ou de la luminosité !

Condensateur



Anologue d'un barrage

Stocke temporairement le courant,
Amorti les grandes variations
de tension



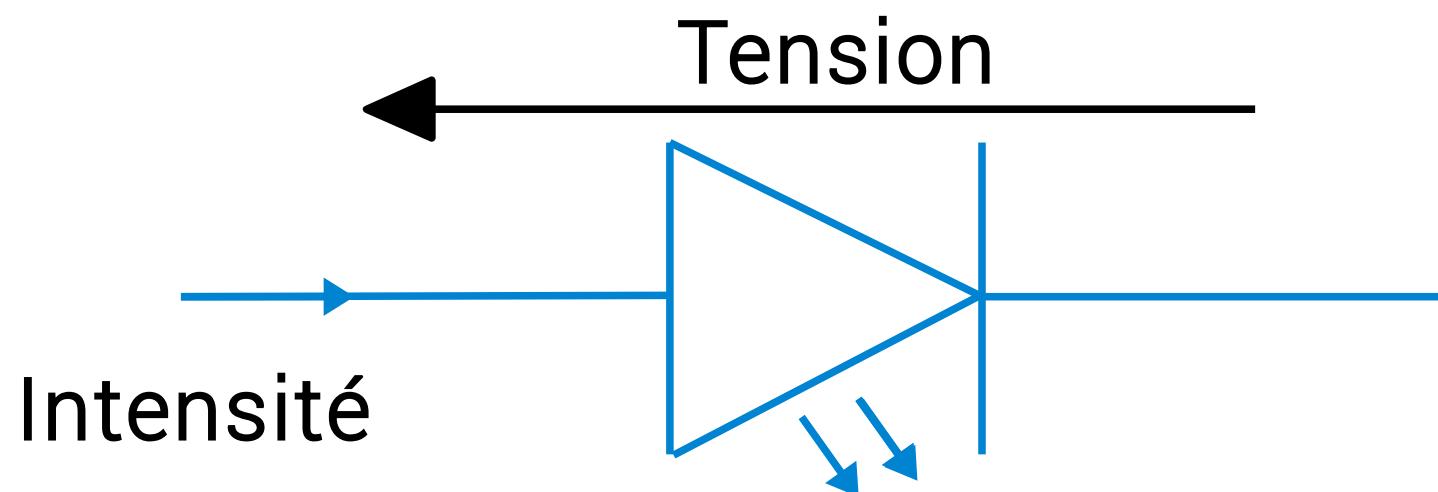
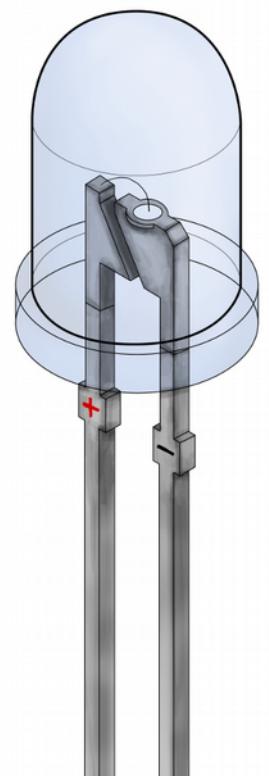
$$\text{Intensité} = C \times \text{Variation de la tension}$$

Diode / LED



Bloque le passage du courant
dans un sens !

Sert comme indicateur lumineux



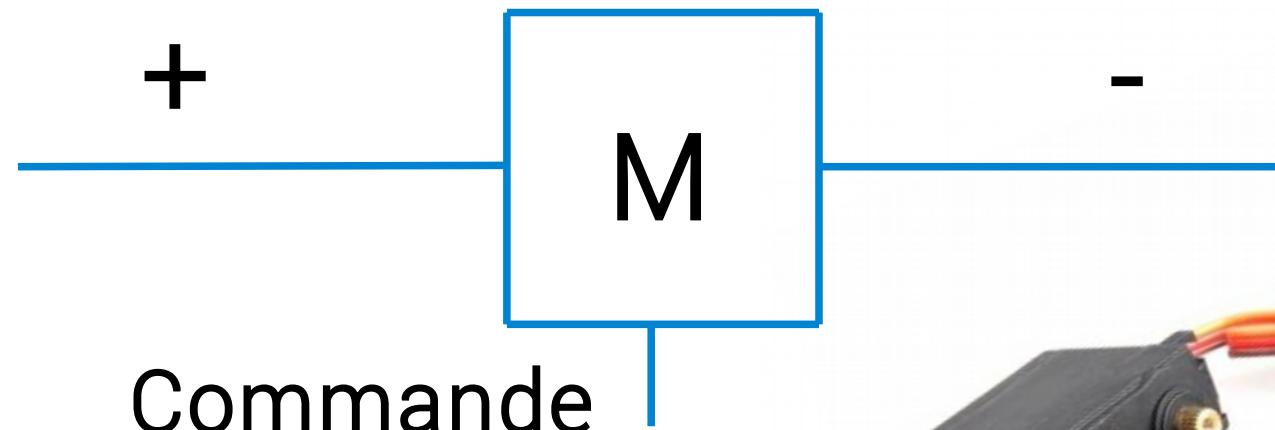


Moteur

Analogue d'un moulin à eau

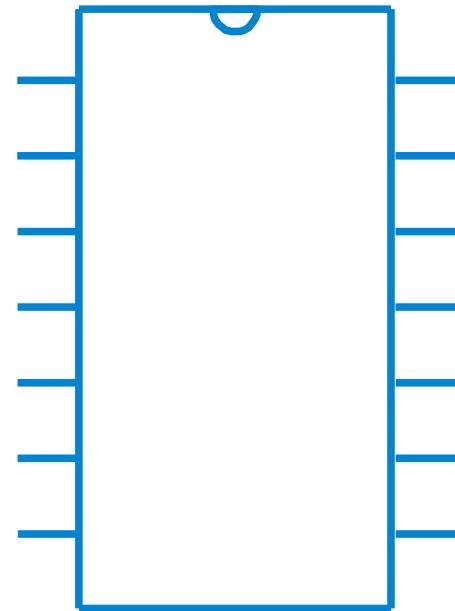


Beaucoup de types de
moteur différents !



Puces électroniques

Énormément de fonctions possibles,
de la simple opération binaire
au microprocesseur...



Analogique, numérique

Deux 'types' d'électroniques,
adaptées à des applications différentes

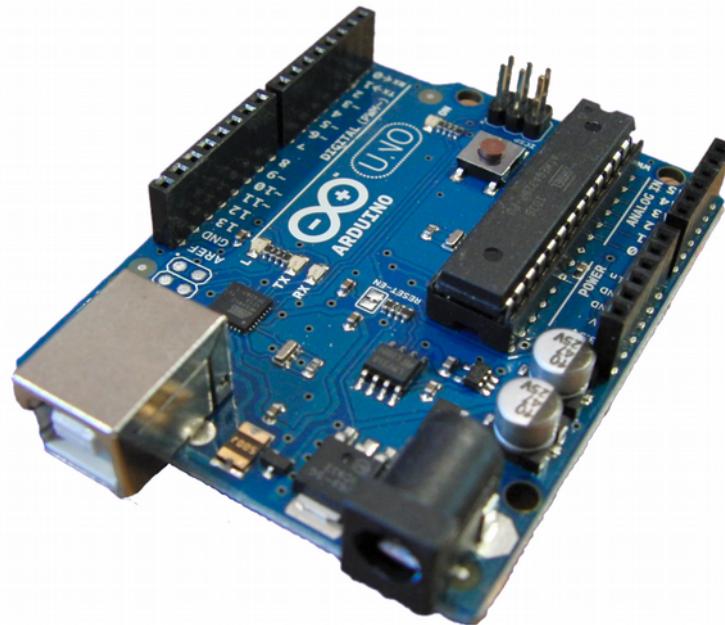
Analogique : tensions arbitraires

capteurs, filtres, traitement
des ondes/signaux (radio, musique, ...)

Numérique : deux états, 0 = "0V" ou 1 = "5V"

interrupteurs, calcul binaire,
transmission de données, ...

Arduino, qu'est-ce que c'est



- C'est
- un circuit imprimé (ou carte électronique)
 - un outil de prototypage
 - ce qui contient l'«intelligence» dans votre montage

Ça n'est pas

un ordinateur (mais ça y ressemble sur le principe)

Mettre en relation des choses...



Afficheurs



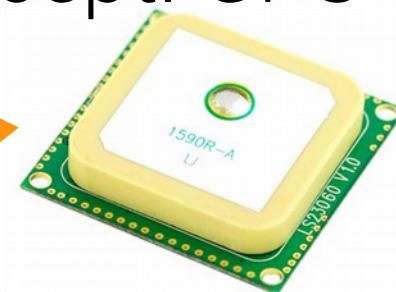
Moteurs



Ordinateur



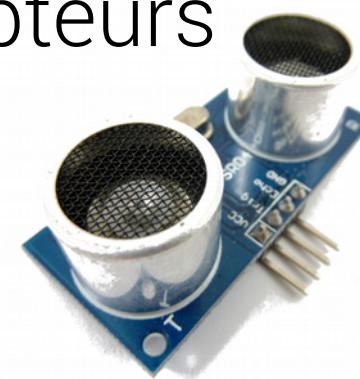
Recept. GPS



Contrôles



Capteurs

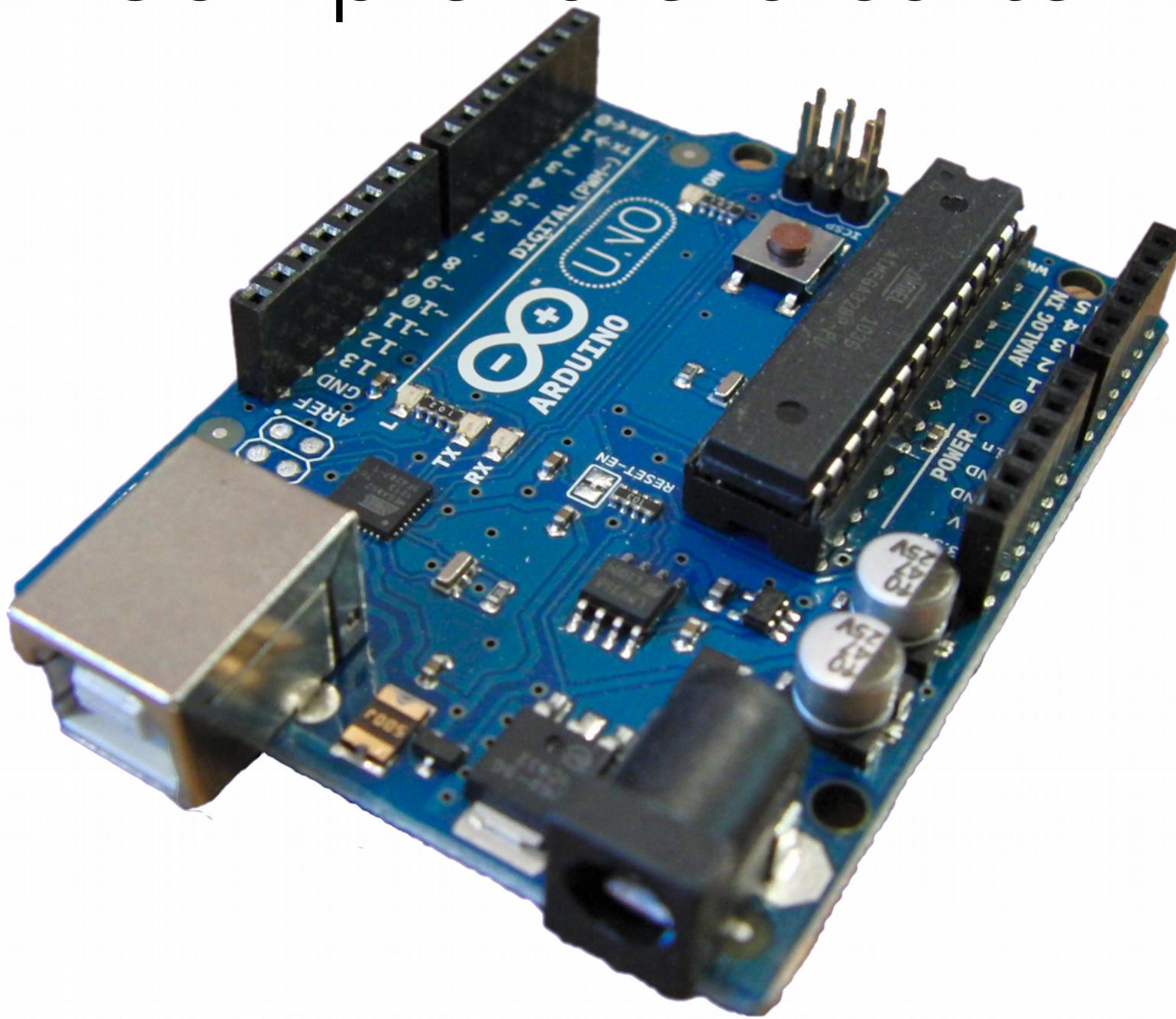


Pourquoi Arduino

et pas autre chose ?

- Simple, flexible
- Pas cher
- Robuste
- Libre / open
- Communauté
- Produits compatibles

Comprendre la carte



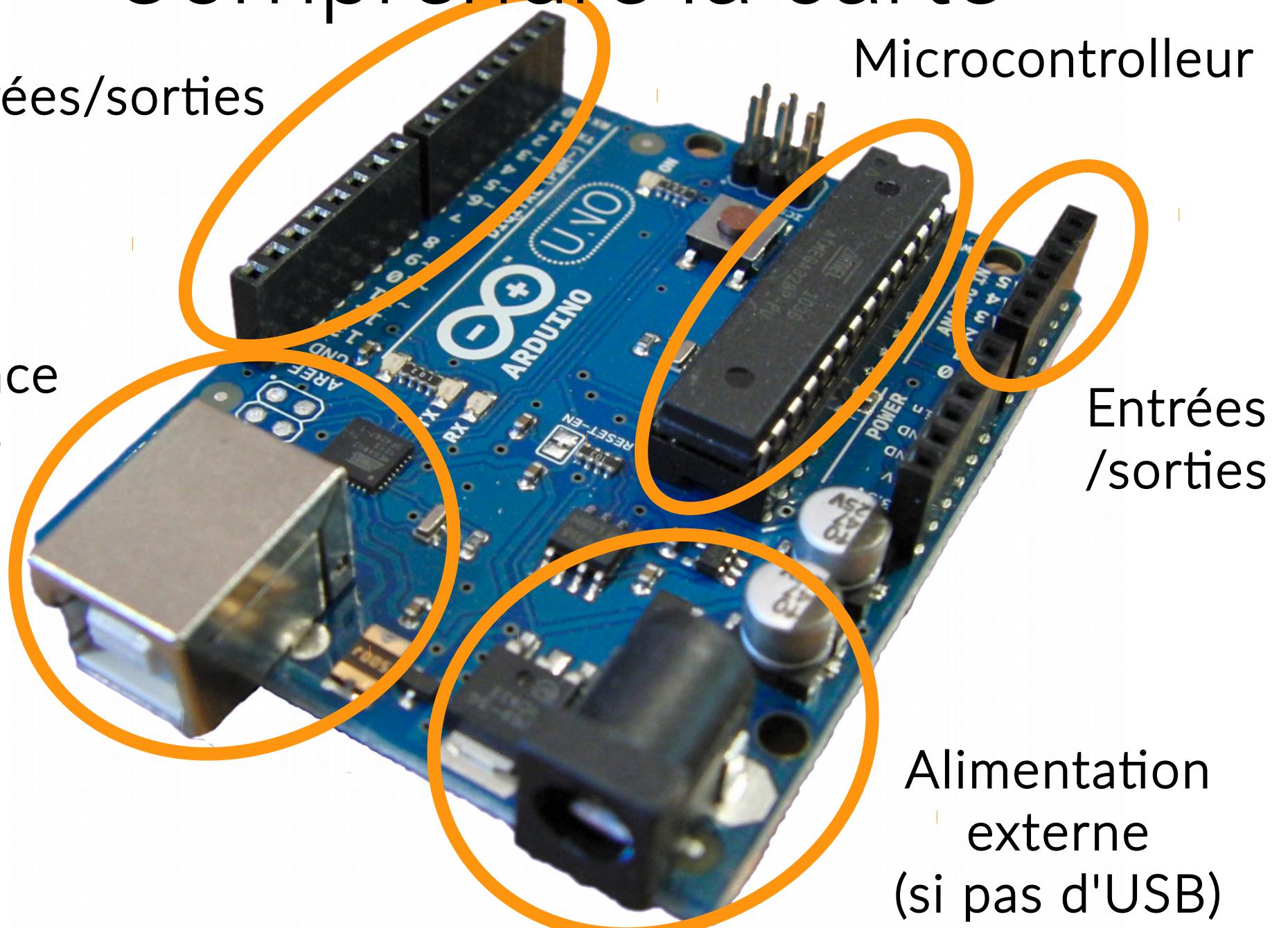
Comprendre la carte

Entrées/sorties

Microcontrôleur

Interface
USB

Entrées
/sorties

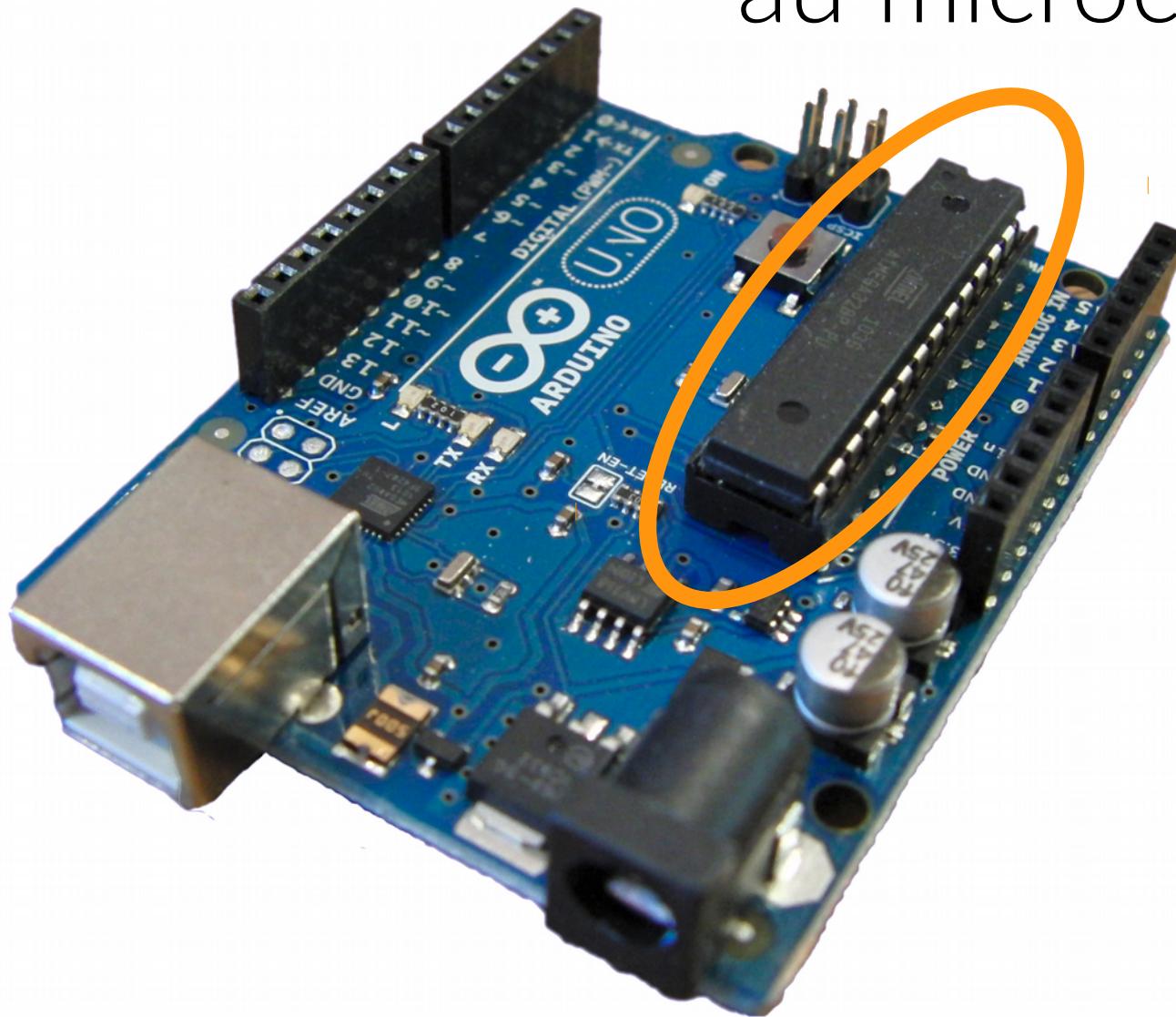


Alimentation
externe
(si pas d'USB)

Mettre en relation,
mais comment ?

Où est l'«intelligence» ?

Ici ! Dans le programme
que vous fournissez
au microcontrôleur !



La programmation

L'art d'expliquer à la machine
ce qu'elle doit faire, sans ambiguïté !

Se fait via un language compréhensible
à la fois par la machine et par l'humain

Une langue ... excepté que votre interlocuteur ne peut
pas comprendre si il y a la moindre erreur de grammaire

Programmer un Arduino



The screenshot shows the Arduino IDE interface with a sketch titled "Blink". The code is written in C++ and follows a standard structure:

```
// Ceci est un commentaire
// Description en français de
// ce que fait votre programme...
//

void setup()
{
    premiereInstruction();
    deuxiemeInstruction();
    ...
}

void loop()
{
    premiereInstruction();
    deuxiemeInstruction();
    ...
}
```

The IDE has a dark-themed interface with a teal header bar. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for file operations like Open, Save, and Upload. The code editor shows the "Blink" sketch with its comments and function definitions. At the bottom of the screen, there is a status bar indicating "Arduino Uno on /dev/ttyACM0".

Programmer un Arduino

The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes File, Edit, Sketch, Tools, Help, and a language switch. Below the menu is a toolbar with various icons. The main area displays the code for the "Blink" sketch:

```
// Ceci est un commentaire
// Description en français de
// ce que fait votre programme...
void setup()
{
    premiereInstruction();
    deuxiemeInstruction();
    ...
}

void loop()
{
    premiereInstruction();
    deuxiemeInstruction();
    ...
}
```

The code uses color-coded syntax highlighting: red for comments, green for the setup() function, blue for the loop() function, and black for the rest of the code.

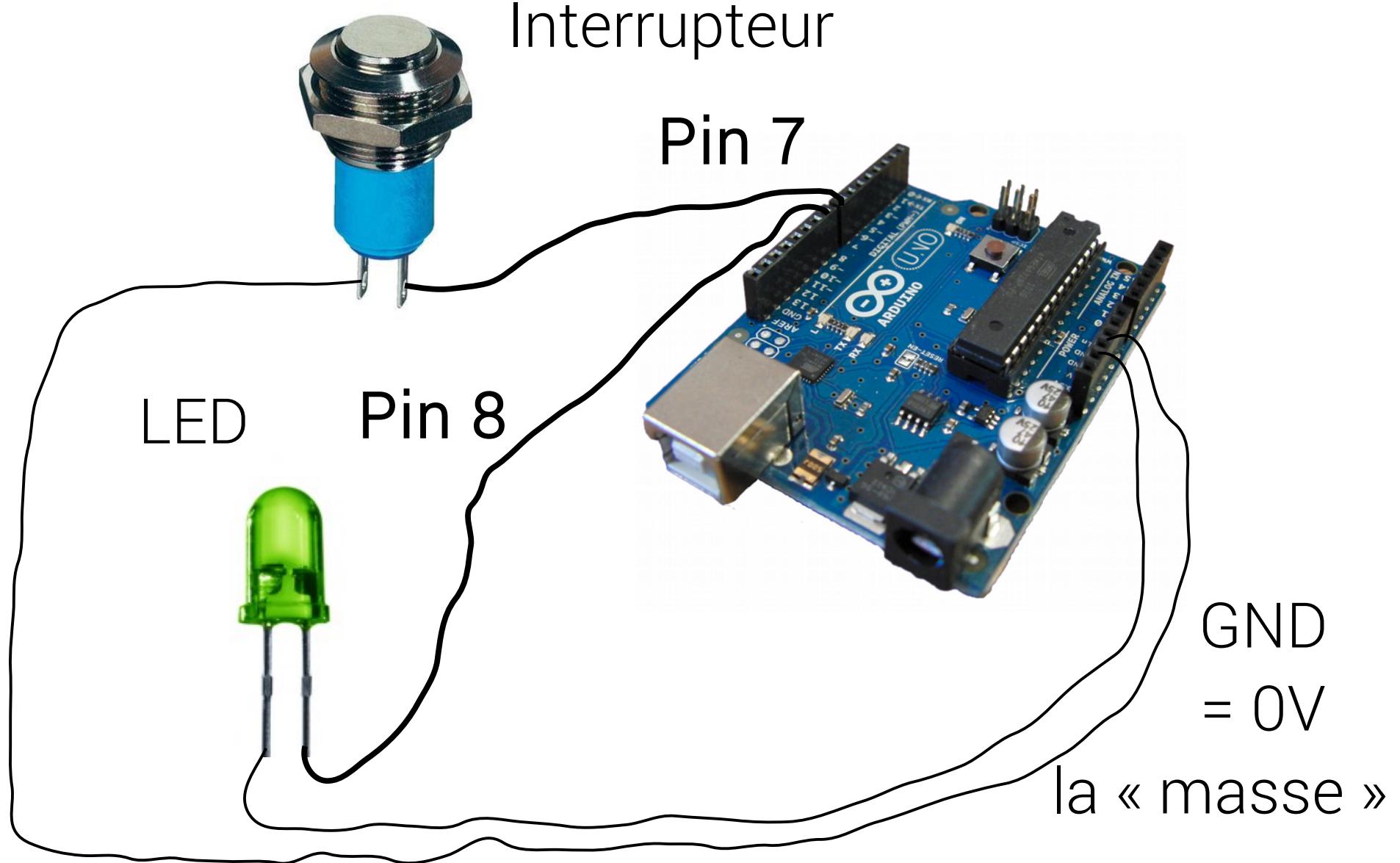
Boutons pour vérifier
le programme et
l'envoyer sur l'Arduino

Commentaires qui
décrivent ce que fait le
programme

Choses à faire lorsque
l'Arduino s'allume

Choses à faire en
boucle par la suite...

Programmer un Arduino



Programmer un Arduino

```
void setup()
{
    pinMode(7, INPUT);
    pinMode(8, OUTPUT);
}

void loop()
{
    int etat = digitalRead(7);

    if (etat == HIGH)
    { digitalWrite(8,HIGH); }

    else
    { digitalWrite(8,LOW); }
}
```

Lorsque l'Arduino s'allume :

- Considérer la pin n°7 comme une entrée (le bouton)
- Considérer la pin n°8 comme une sortie (la LED)

Ensuite, faire en boucle :

- Vérifier l'état du bouton (pin 7)
- Si l'état est 1,
allumer la LED (pin 8 à 1)
- Sinon,
éteindre la LED (pin 8 à 0)

Programmer un Arduino

```
#define PIN_BOUTON 7
#define PIN_LED     8

void setup()
{
    pinMode(PIN_BOUTON, INPUT );
    pinMode(PIN_LED,    OUTPUT);
}

void loop()
{
    int etat = digitalRead(PIN_BOUTON);

    if (etat == HIGH)
    { digitalWrite(PIN_LED,HIGH); }

    else
    { digitalWrite(PIN_LED,LOW); }
}
```

Programmer un Arduino

```
#define PIN_BOUTON 7
#define PIN_LED     8

void setup()
{
    pinMode(PIN_BOUTON, INPUT );
    pinMode(PIN_LED,      OUTPUT);
}

void loop()
{
    int etat = digitalRead(PIN_BOUTON);

    if (etat == HIGH) allumerLED();
    else              eteindreLED();
}

void allumerLED() { digitalWrite(PIN_LED,HIGH); }

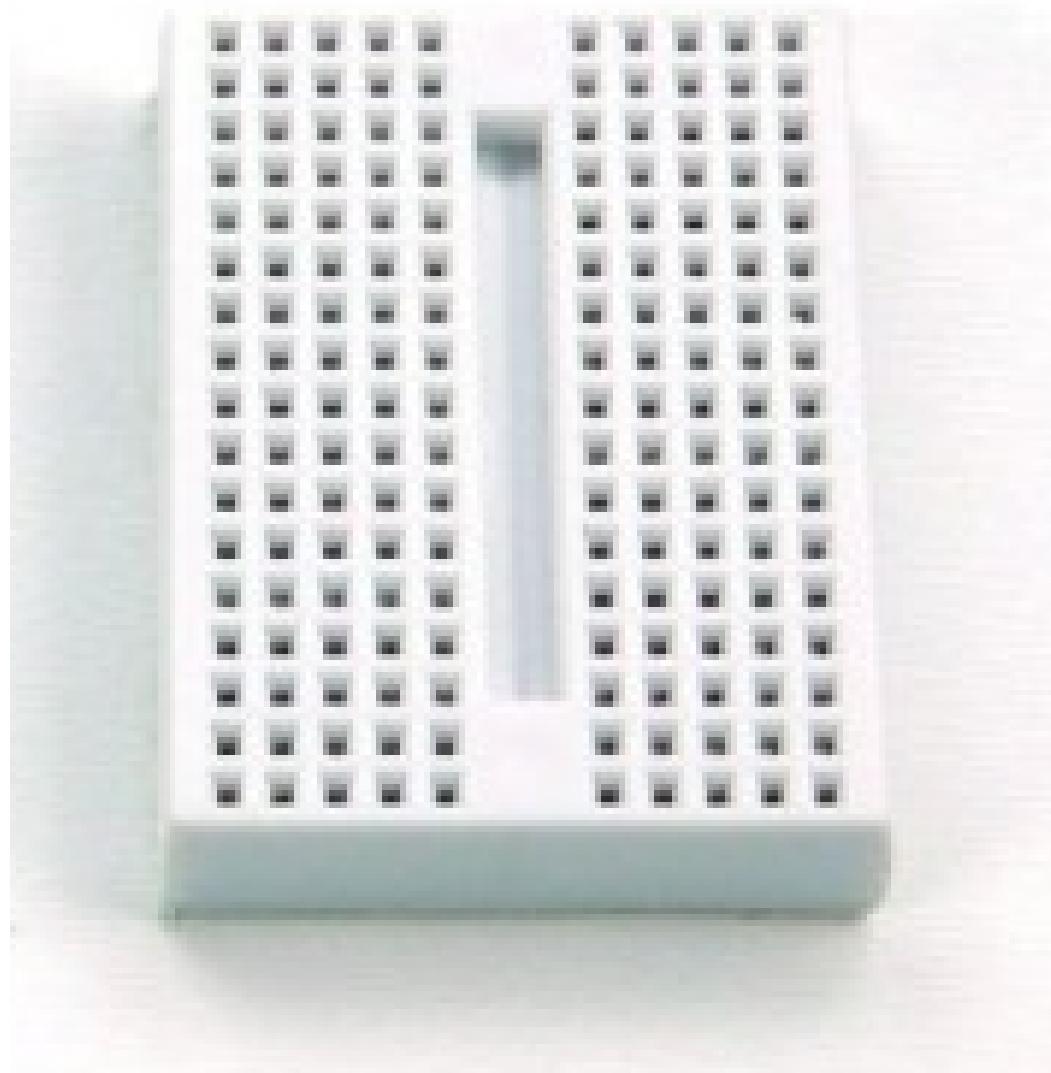
void eteindreLED() { digitalWrite(PIN_LED,LOW); }
```

(2)

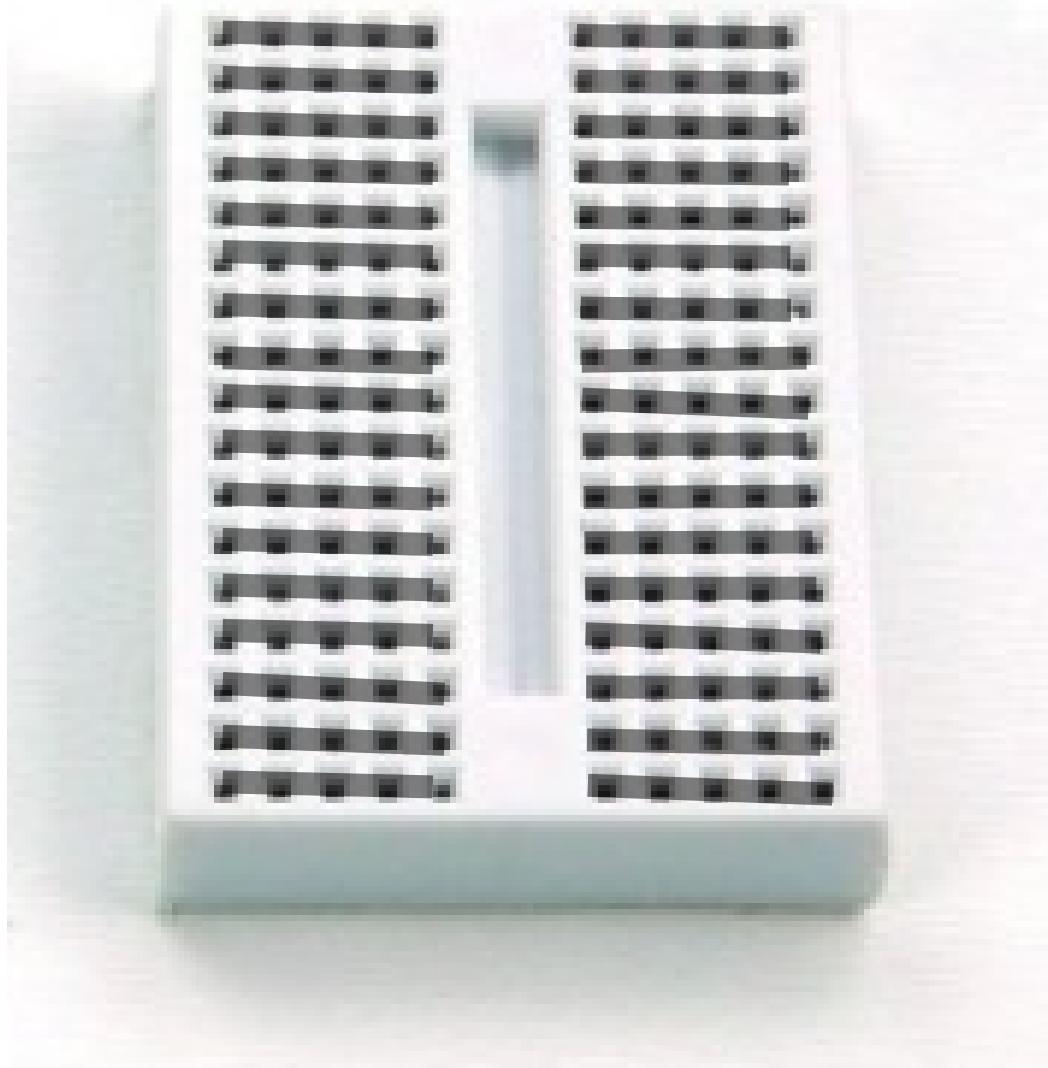
Votre premier montage Arduino

- Protéger une LED
 - Le montage
- Le mini-breadboard
 - À vous !

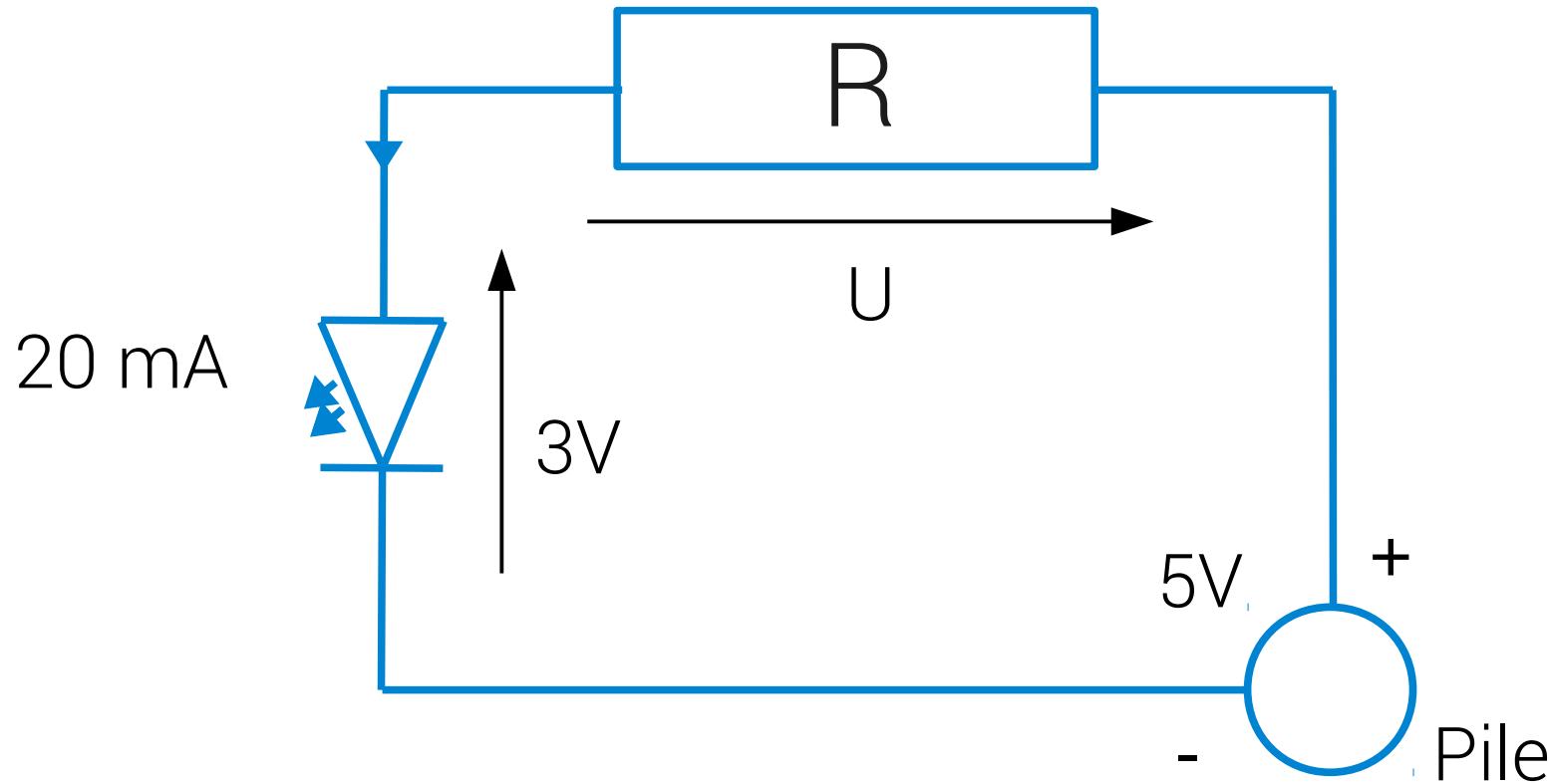
Le mini-breadboard



Le mini-breadboard



Protéger une LED



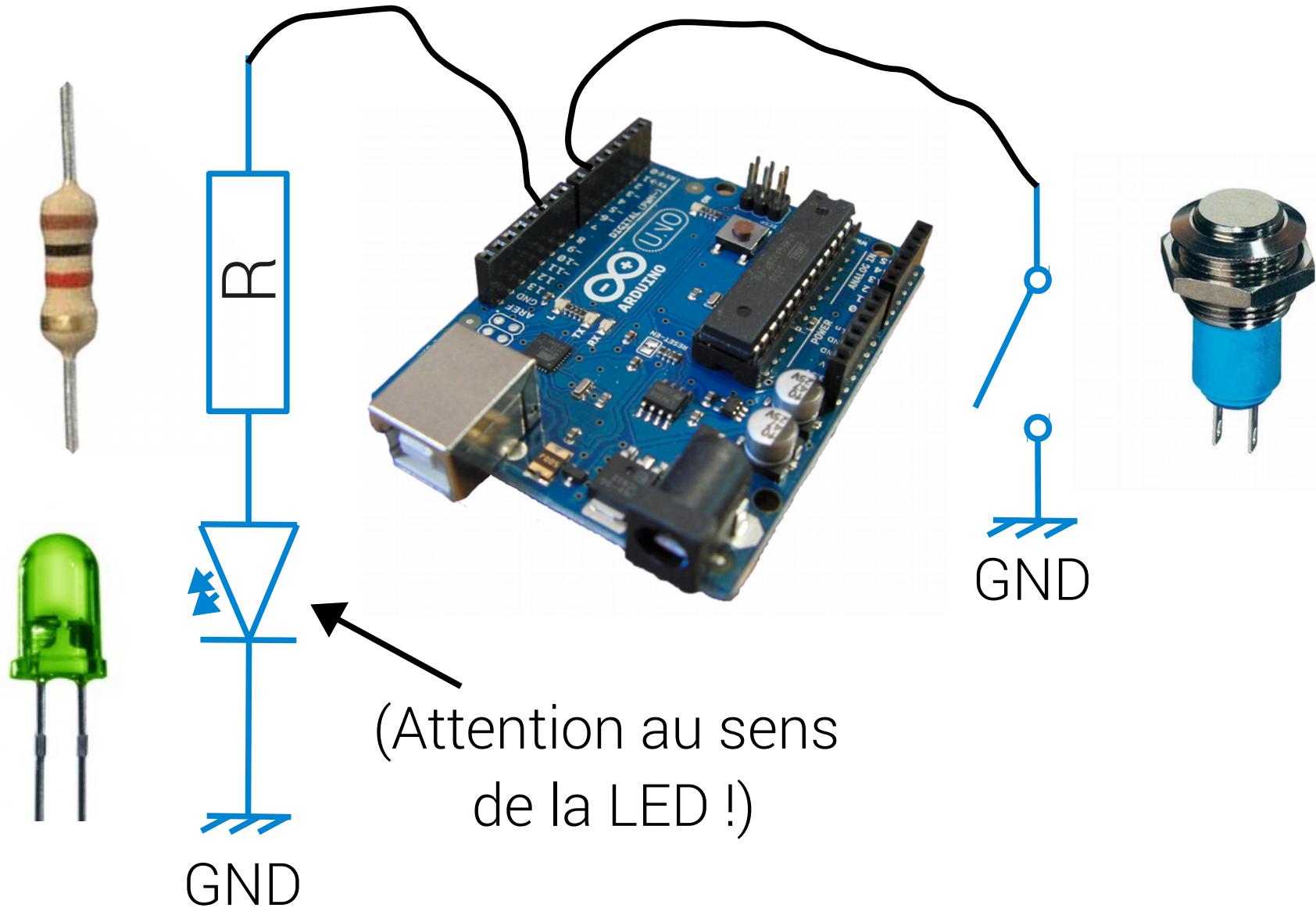
$$5V = 3V + U$$

$$\rightarrow U = 2V$$

$$\text{Loi d'Ohm : } U = Ri$$

$$\rightarrow R = U / i = 2V / 20 \text{ mA} = 100\Omega$$

Le montage



Le programme

```
#define PIN_BOUTON 7
#define PIN_LED     8

void setup()
{
    pinMode(PIN_BOUTON, INPUT );
    pinMode(PIN_LED,   OUTPUT);
}

void loop()
{
    int etat = digitalRead(PIN_BOUTON);

    if (etat == HIGH) allumerLED();
    else              eteindreLED();
}

void allumerLED() { digitalWrite(PIN_LED,HIGH); }

void eteindreLED() { digitalWrite(PIN_LED,LOW); }
```

À vous ! 😊

(3)

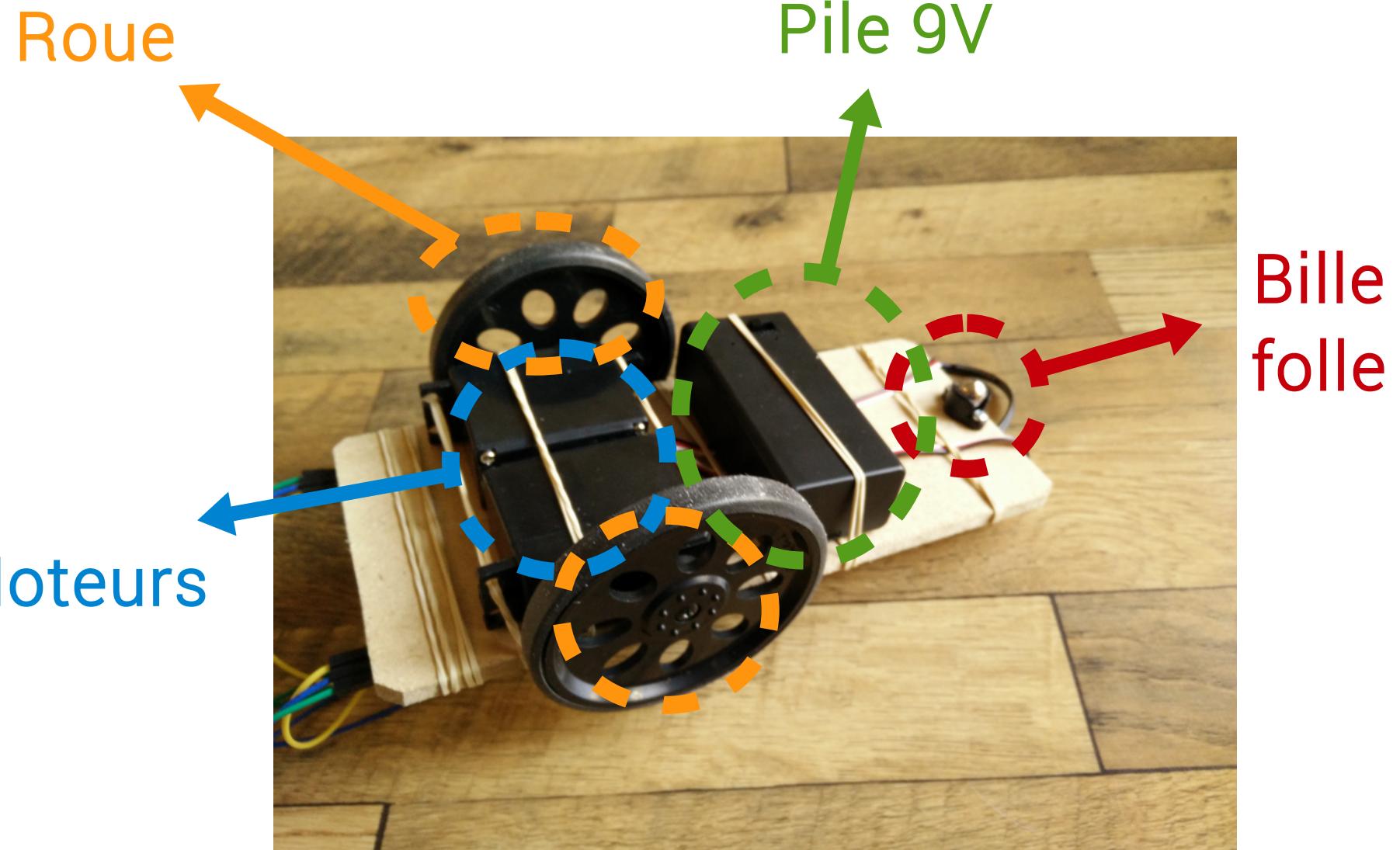
Construction & programmation d'un robot

- Les etapes de construction
 - Le programme
 - A vous !

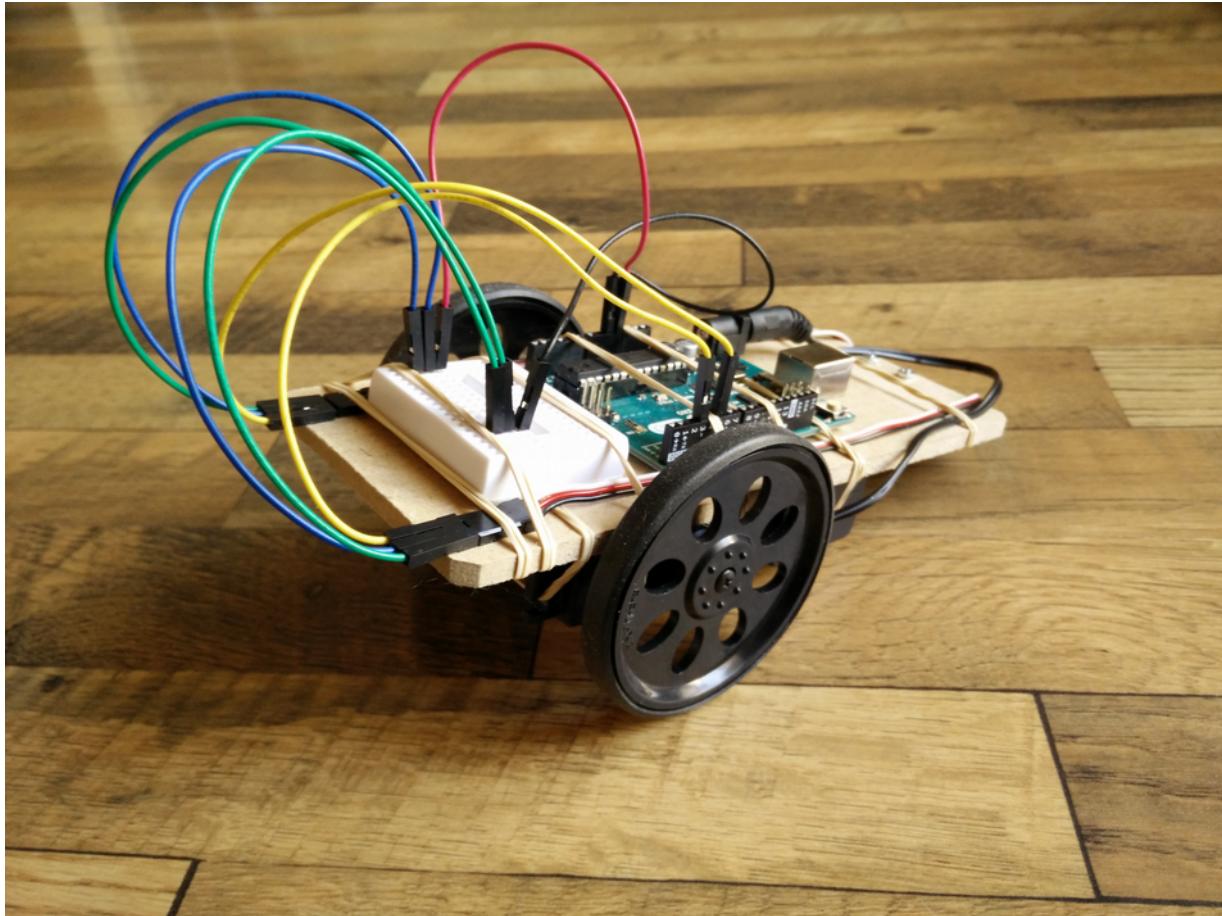
Les etapes de construction



Les etapes de construction

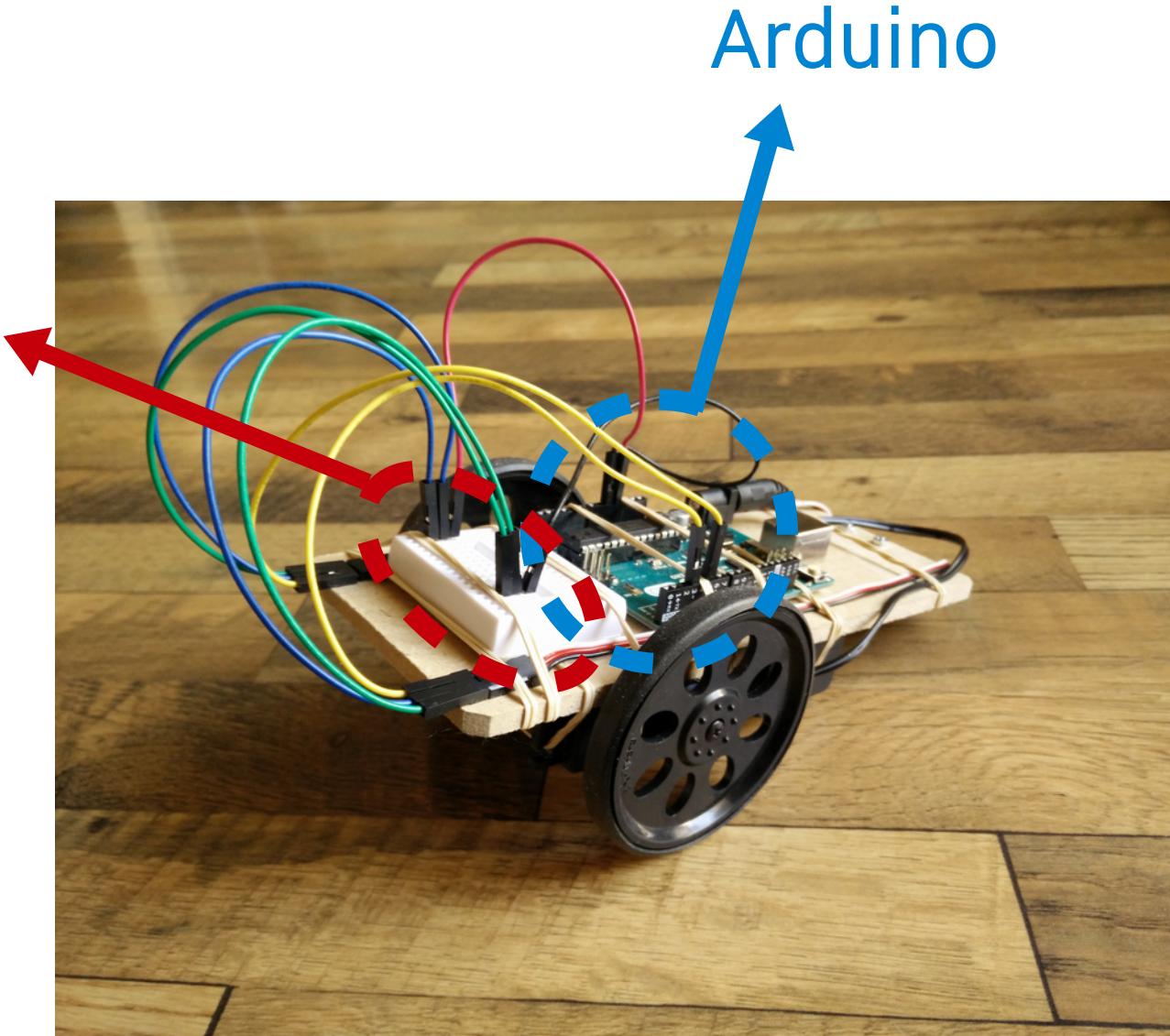


Les etapes de construction

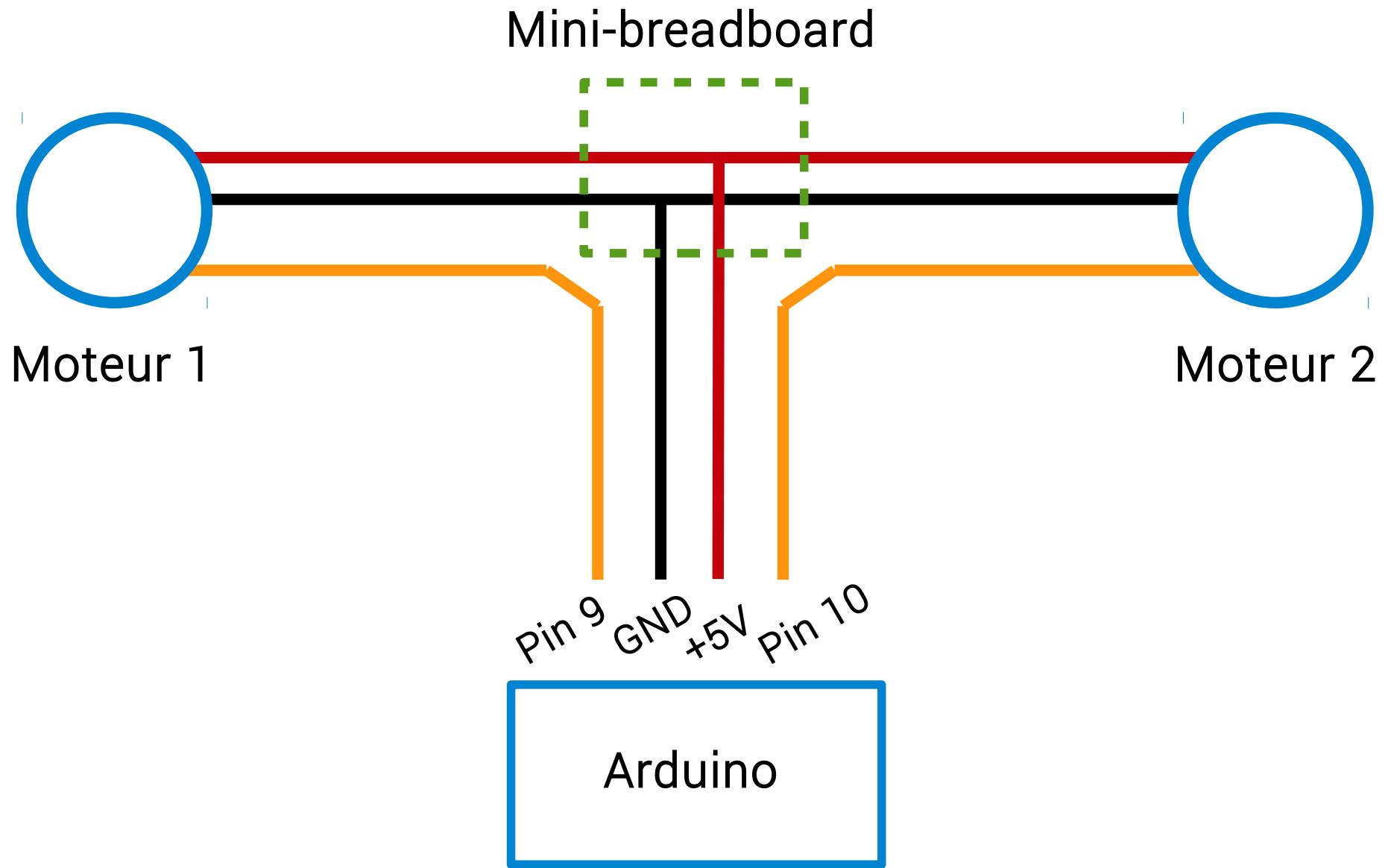


Les etapes de construction

Mini
Bread
board



Les etapes de construction



Le programme

```
void setup()
{
}

void loop()
{
    avancer(5,2000);

    delay(1000);

    tourner(0.25);

    delay(1000);

    //calibrationRepos();
    //calibrationRotation();
}
```

Faire en boucle :

- Avancer à vitesse 5 pendant 2sec
 - Attendre 1sec
 - Tourner d'un quart de tour (0.25)
 - Attendre 1sec
- fonctions pour la calibration du robot

À vous ! 😊