

# Vittascience

## Vittascience FR

Vittascience : Une plateforme innovante pour enseigner la programmation

Vittascience est une plateforme pédagogique en ligne qui permet d'enseigner la programmation informatique et l'intelligence artificielle aux enfants de manière ludique et interactive. Elle propose une interface intuitive où les enfants peuvent coder en blocs, à la manière de Scratch, ou en code, pour ceux qui souhaitent progresser vers des langages plus complexes. L'interface en mode hybride, comprenant les modes blocs et code juxtaposés, permet de mieux comprendre le lien entre les deux modes de langage. Vittascience offre une transition en douceur entre ces deux modes, permettant aux enfants de développer leurs compétences à leur propre rythme.

cataloguevittascience2025.pdf

## Programmer un arduino

### Vittascience et Arduino



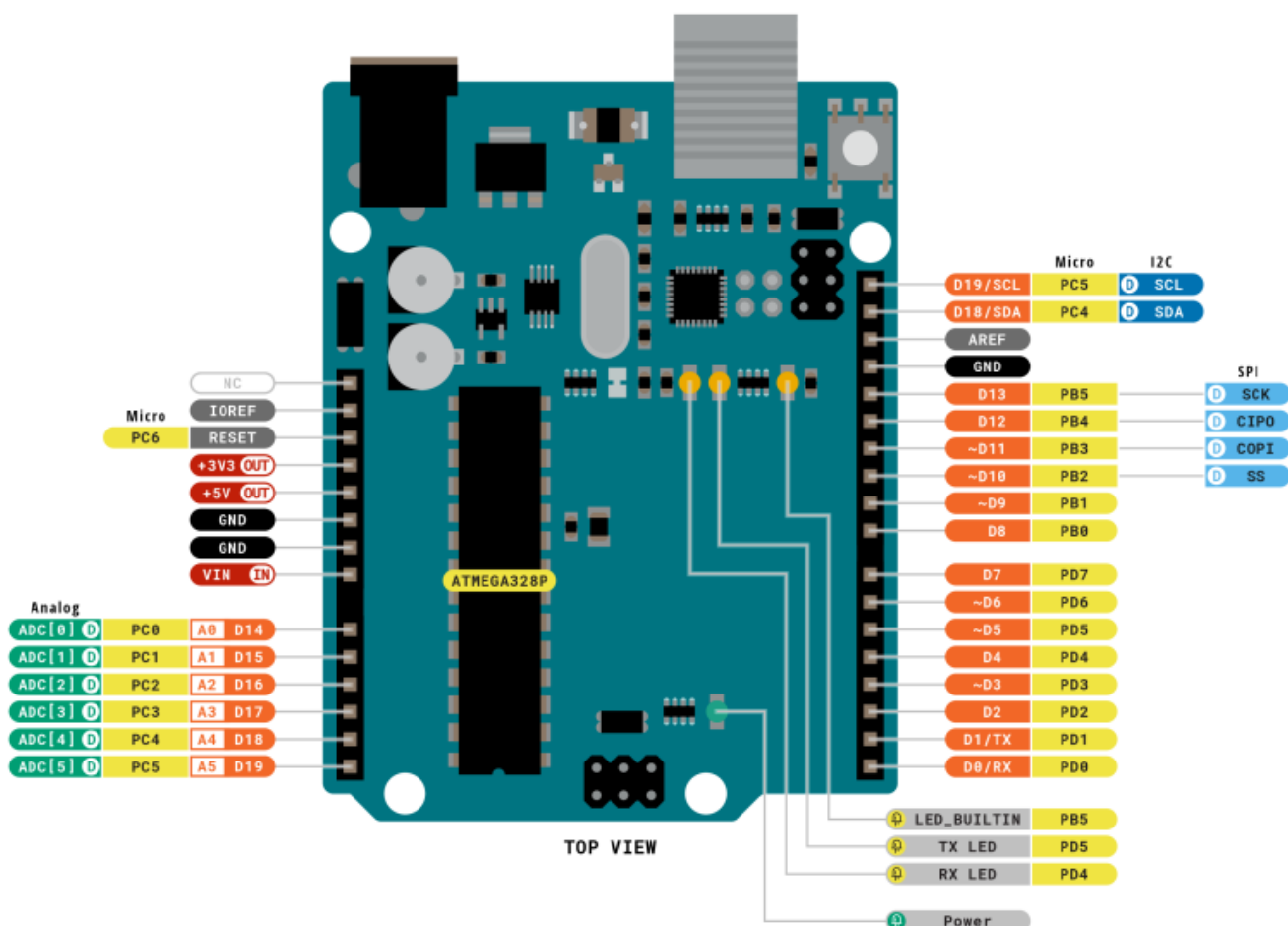
## Decouverte Arduino

### Manipulation de la carte Arduino Uno et de sa LED intégrée



## Decouverte Arduino UNO

[Prise en main de la carte Arduino UNO et de l'interface Vittascience](#)



<b>Legend:</b>	<span style="color: orange;">■</span> Digital	<span style="color: blue;">■</span> I2C
<span style="color: red;">■</span> Power	<span style="border: 1px solid black; display: inline-block; width: 10px; height: 10px;"></span> Analog	<span style="color: lightblue;">■</span> SPI
<span style="background-color: black; color: black;">■</span> Ground	<span style="background-color: yellow;">■</span> Main Part	<span style="color: green;">■</span> Analog

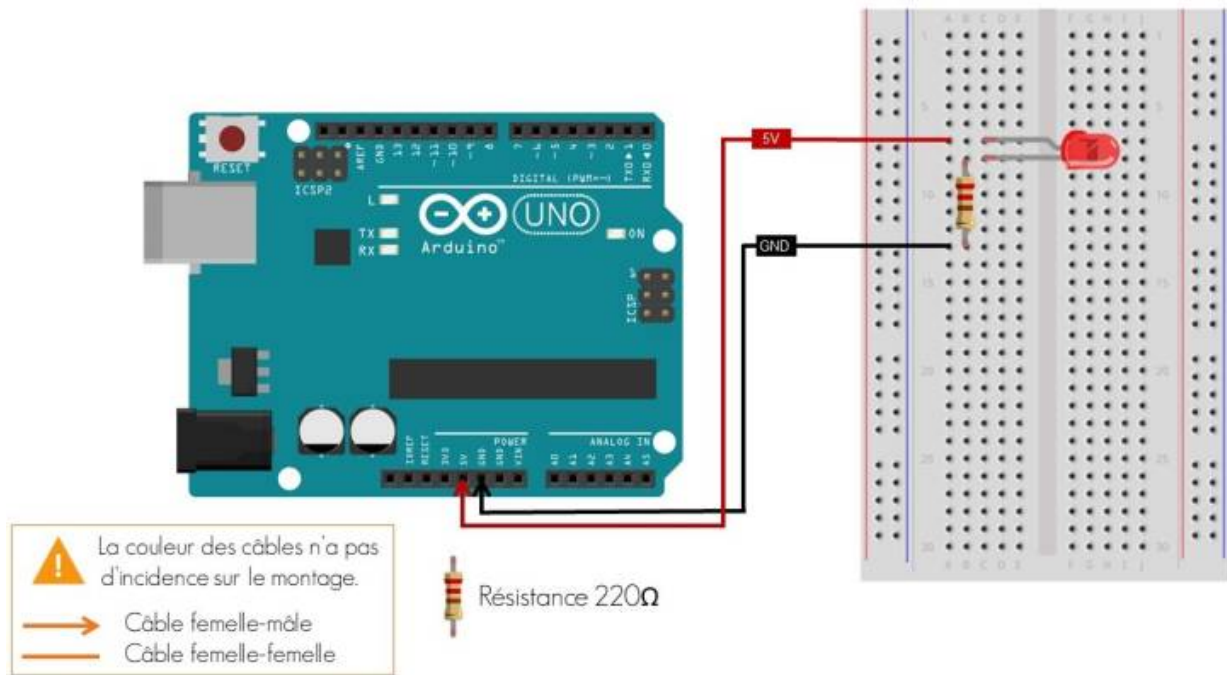


ARDUINO  
 ARDUINO UNO REV3  
 SKU code: A000066  
 Pinout  
 Last update: 6 Oct, 2022

prise\_en\_main\_de\_la\_carte\_arduino\_uno\_et\_de\_l\_interface\_vittascience.pdf

## Decouverte LED et Arduino

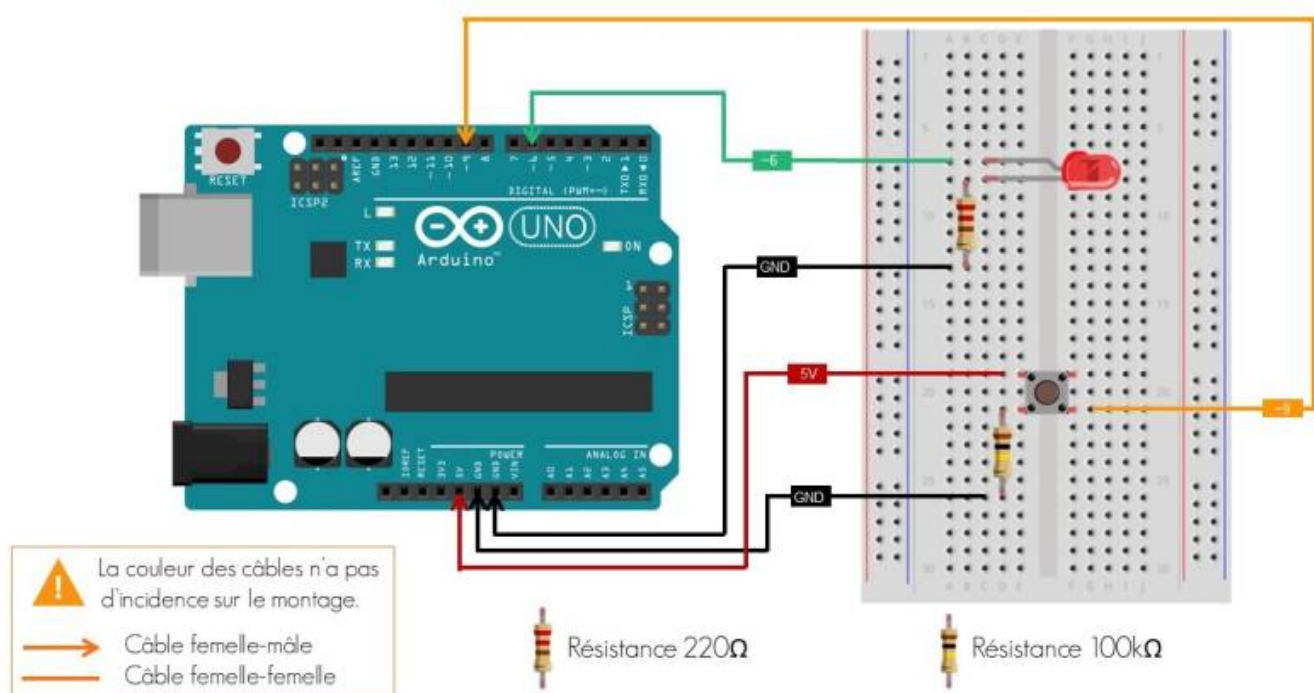
[Allumer une LED avec Arduino](#)



allumer\_une\_led\_avec\_arduino.pdf

## Decouverte Bouton + LED + Arduino

Ajouter un bouton sur Arduino



ajouter\_un\_bouton\_sur\_arduino.pdf

## Decouverte PhotoResistance + Led + Arduino

Mesurer la luminosité avec une photorésistance sur Arduino



[mesurer\\_la\\_luminosite\\_avec\\_un\\_photoresistance\\_sur\\_arduino.pdf](#)

## Decouverte Vitesse du son avec un HC-SR04 + Arduino

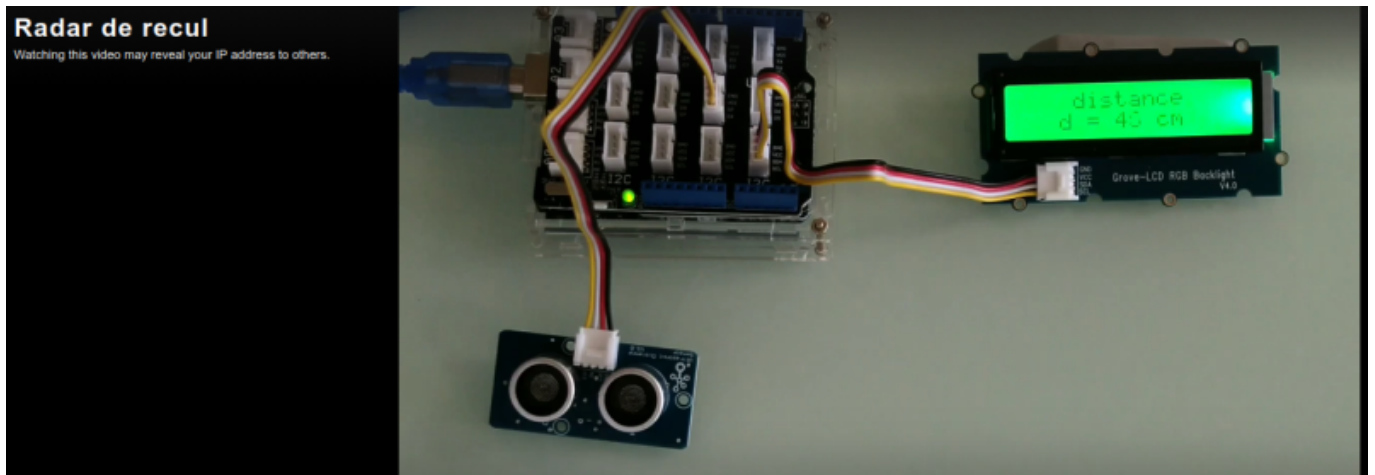
Vitesse du son : Capteur ultrason HC-SR04



[vitesse\\_du\\_son.pdf](#)

## Decouverte Radar de recul avec HC-SR04 + Arduino

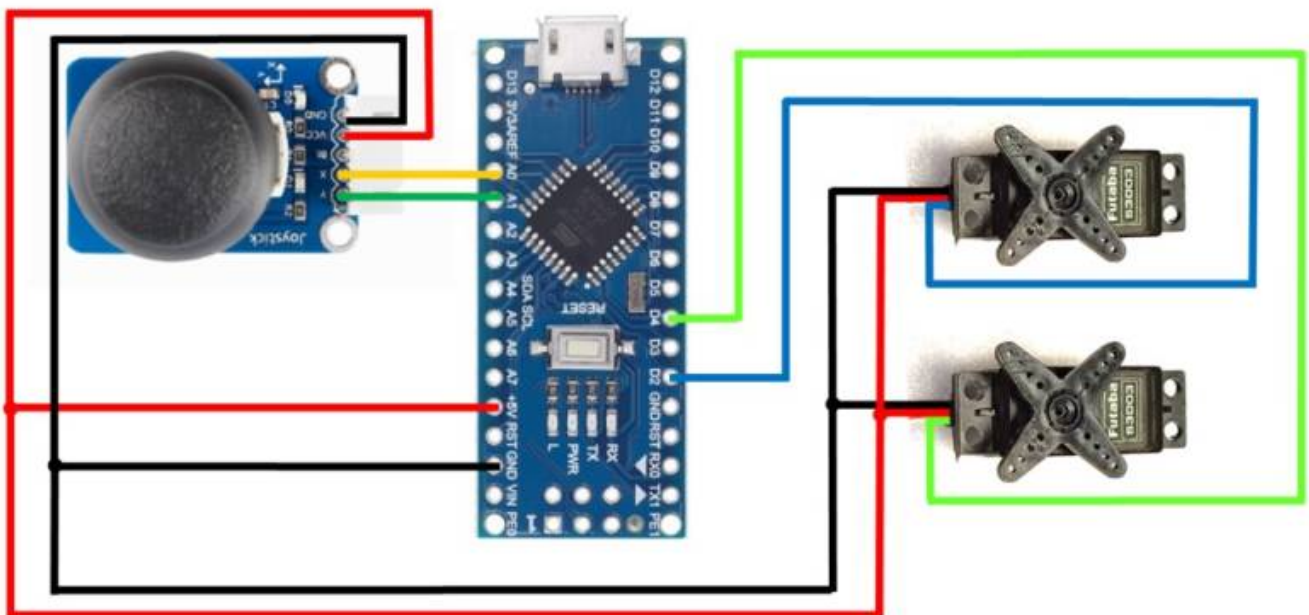
Radar de recul avec un écran LCD



radar\_de\_recul\_avec\_un\_ecran\_lcd\_colore.pdf

## Decouverte Chariot+ servo Moteur 360° + Arduino

Chariot téléguidé par un joystick



chariot\_teleguide\_par\_un\_joystick.pdf

# Programmer un Robot Mbot

## Televerser dans le robot mBot1 à partir de l'interface Vittascience

Pour Téléverser le programme robot sur mBot1 à partir de l'interface Vittascience ( si "Televerser" ne fonctionne pas , passer par les étapes 1 et 2 ... ) :

1- [Vider le cache de Google Chrome](#)

2- [Passer en navigation privée sur Google chrome](#)

3- cliquer sur "Téléverser" ( connecter sur USB et allumer le mBot1 avant ...)

4- choisir le port COM du mBot1 et faire connexion ( exemple: "**USB Serial (COM3)**" sous Windows 11 ou "**USB Serial (ttyUSB0)**" sous Linux)

et c'est tout , dans la console série de vittascience tu dois voir :

```
Le programme utilise 8469 octets d'espace de stockage. (28%)  
Compilation réussie.  
Téléversement du programme ...  
Carte détectée: Arduino Nano w/ ATmega328  
Programme Arduino envoyé avec succès dans la carte !"
```

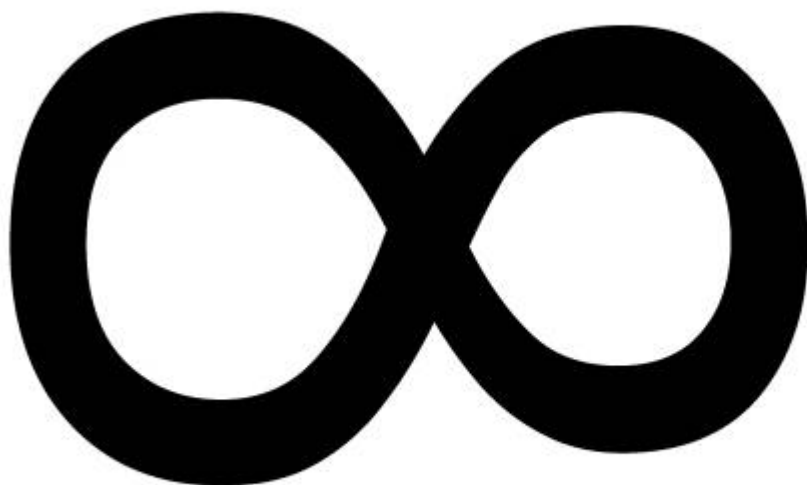
## Exemples Vittascience Robot mBot1

[Mbot et Vittascience](#)

[activitesvittascience.zip](#)

**Image fond ecran pour virtualisation mBot1 sur Vittascience**

[tracrobot004.jpg.zip](#)



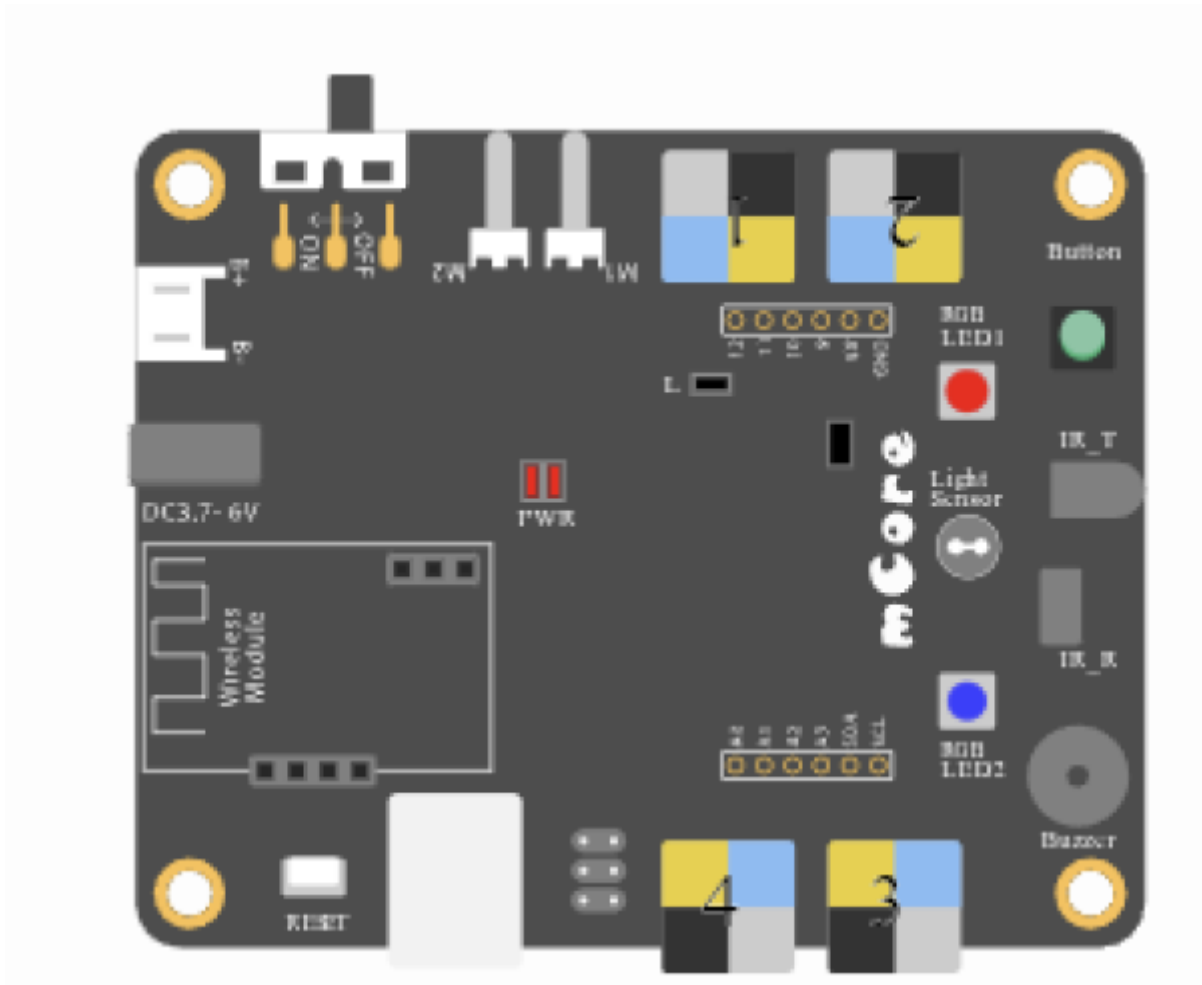
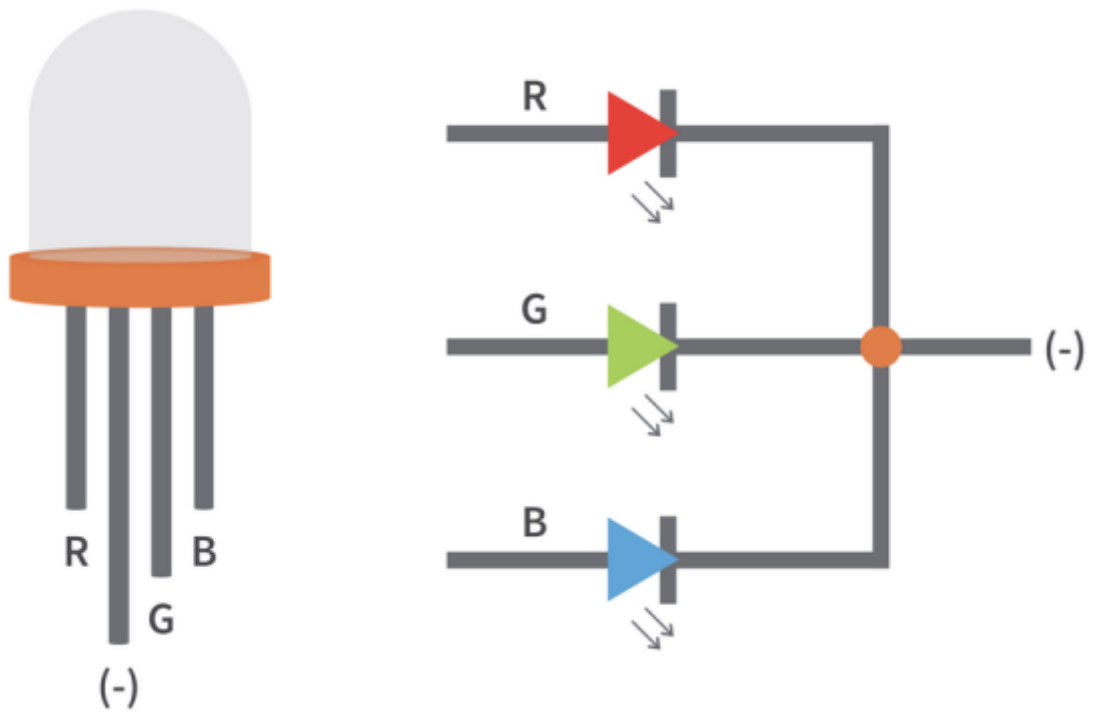
## Programmes mBot1 Vittascience

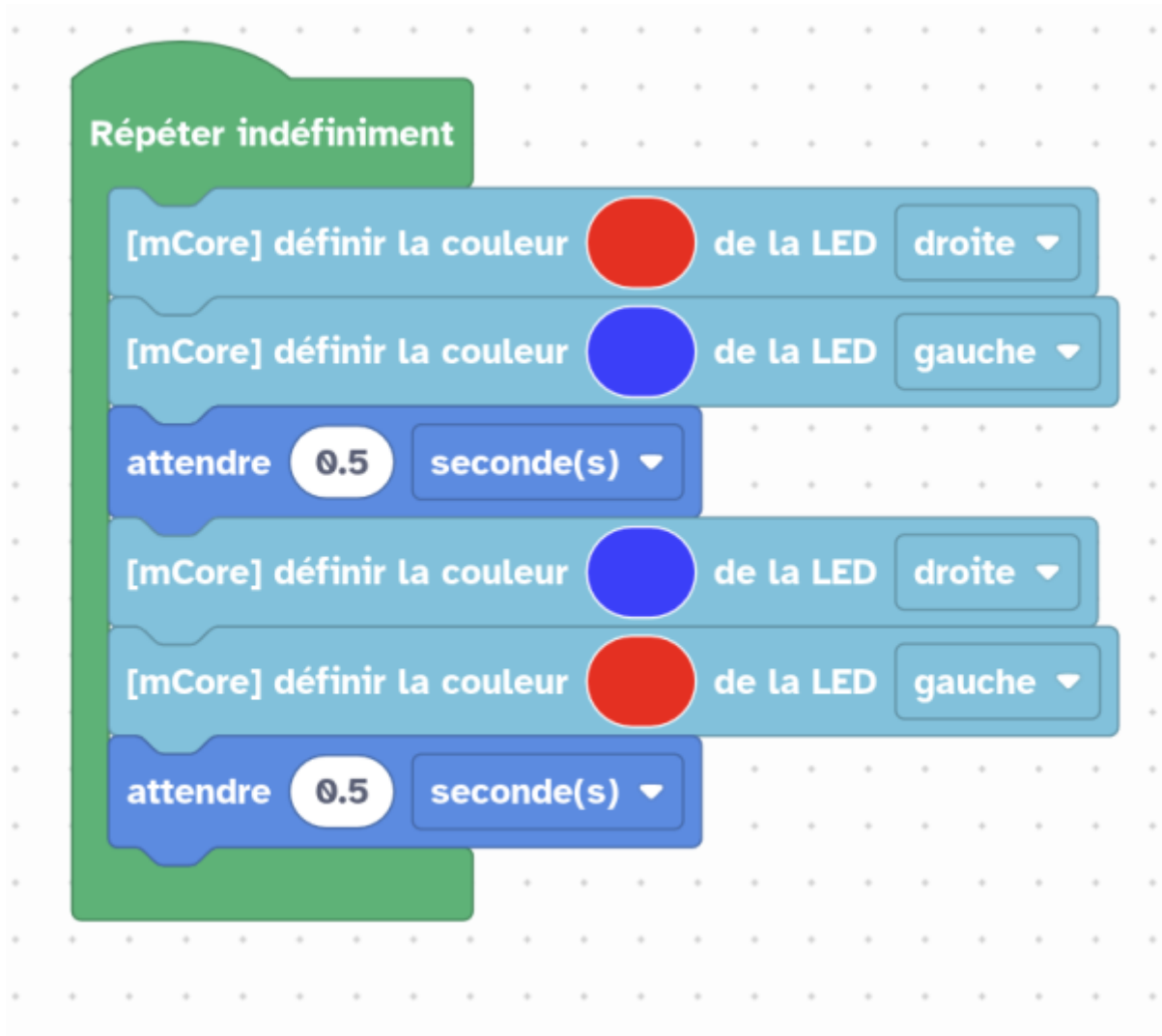
**robot-mbot-utiliser-les-led-rgb**

[robot-mbot-utiliser-les-led-rgb](#)

robot\_mbot\_-\_utiliser\_les\_led\_rgb.pdf







## rgb.ino

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

MeRGBLed rgbled_board(7, 2);

void setup() {
}

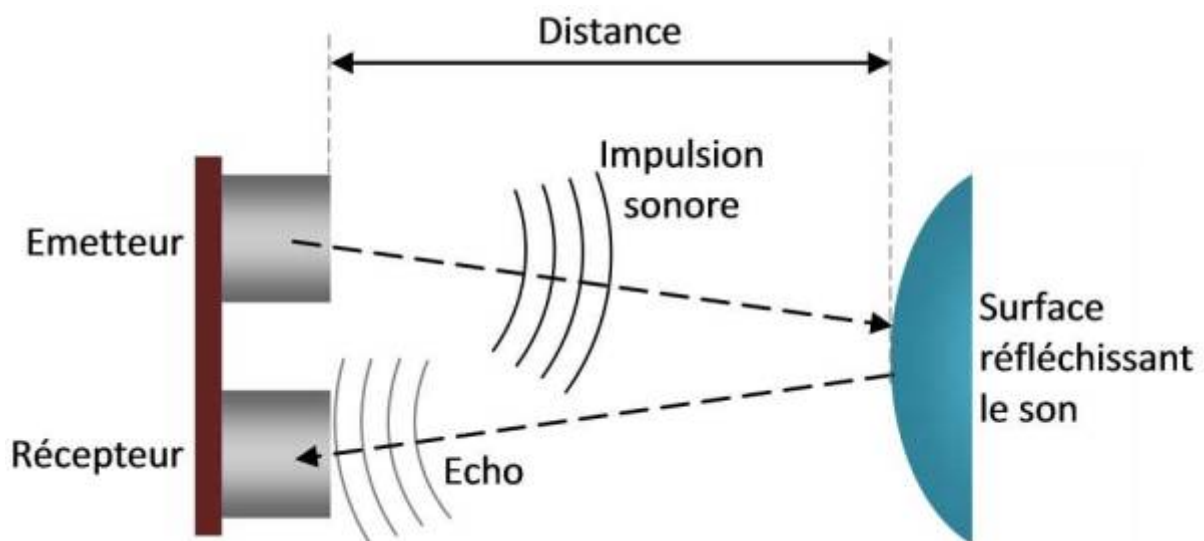
void loop() {
  rgbled_board.setColor(1, 255, 0, 0);
  rgbled_board.show();
  rgbled_board.setColor(2, 51, 51, 255);
  rgbled_board.show();
  delay(1000*0.5);
  rgbled_board.setColor(1, 51, 51, 255);
  rgbled_board.show();
  rgbled_board.setColor(2, 255, 0, 0);
```

```
rgbled_board.show();  
delay(1000*0.5);  
}
```

## Robot mBot - Utiliser le capteur de distance

[robot-mbot-utiliser-le-capteur-de-distance](#)

[robot\\_mbot\\_-\\_utiliser\\_le\\_capteur\\_de\\_distance.pdf](#)





robot\_mbot\_-\_utiliser\_le\_capteur\_de\_distance\_-\_partie\_3\_20251027\_123737.ino.tar  
(Enlever .tar)

[distancearduino.ino](#)

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

// Ultrasonic on PORT_3
MeUltrasonicSensor ultrasonic_3(PORT_3);
MeDCMotor motor_L(9);
MeDCMotor motor_R(10);

void mBot_setMotorLeft(int8_t dir, int16_t speed) {
  speed = speed/100.0*255;
  motor_L.run((9) == M1 ? -(dir*speed) : (dir*speed));
}

void mBot_setMotorRight(int8_t dir, int16_t speed) {
  speed = speed/100.0*255;
  motor_R.run((10) == M1 ? -(dir*speed) : (dir*speed));
}

void setup() {
}

void loop() {
  if (ultrasonic_3.distanceCm() > 5) {
```

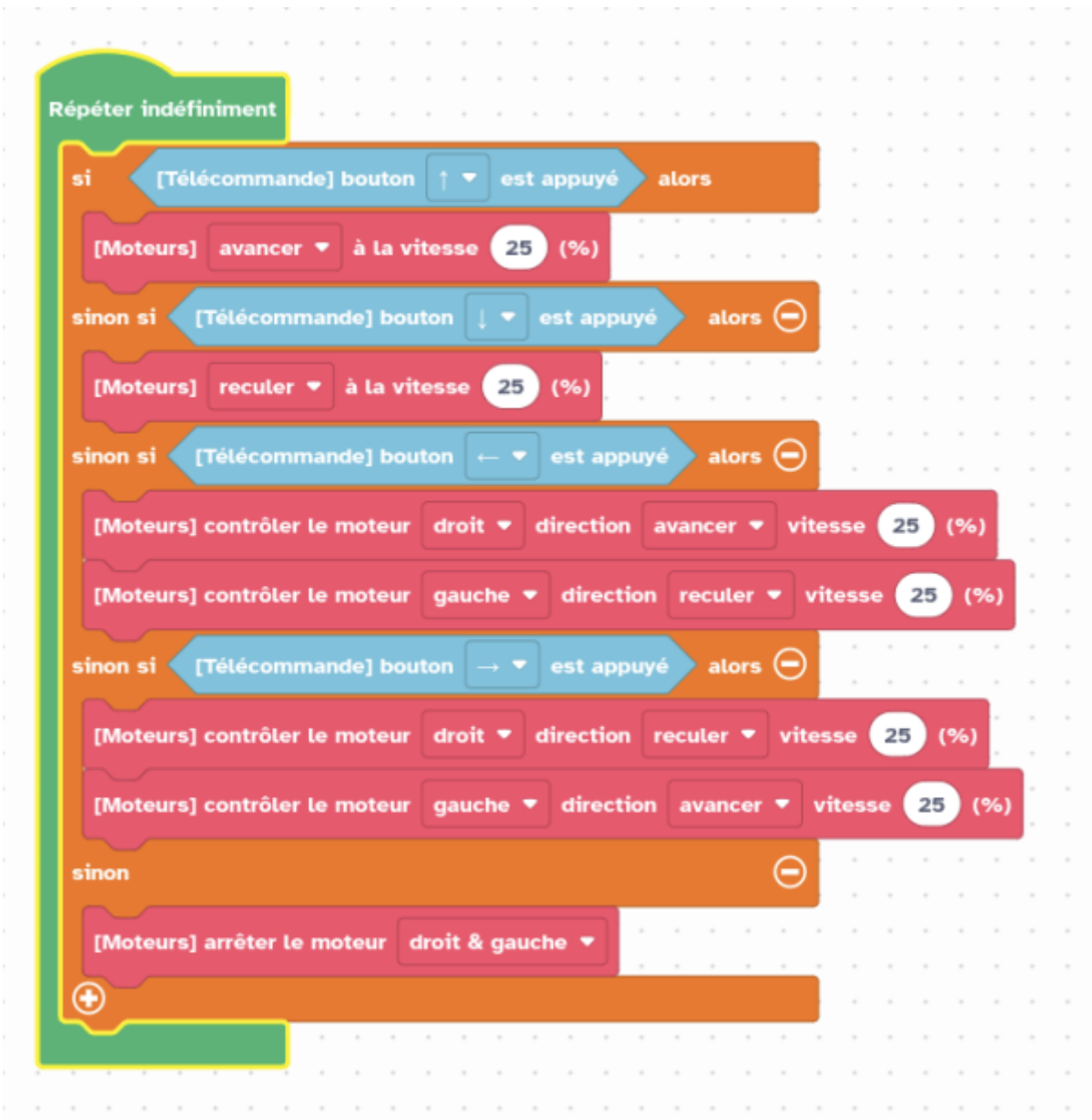
```
mBot_setMotorRight(1, 25);  
mBot_setMotorLeft(1, 25);  
} else {  
  mBot_setMotorRight(0, 0);  
  mBot_setMotorLeft(0, 0);  
}  
}
```

## robot-mbot-controler-le-robot-avec-une-telecommande

[robot-mbot-controler-le-robot-avec-une-telecommande](#)

robot\_mbot\_-\_controler\_le\_robot\_avec\_une\_telecommande.pdf





robot\_mbot\_-\_controler\_le\_robot\_avec\_une\_telecommande\_-\_partie\_3\_20251027\_144323.ino.tar  
( enlever .tar)

[telecommandemBot.ino](#)

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

MeIR ir;
MeDCMotor motor_L(9);
MeDCMotor motor_R(10);

void mBot_setMotorLeft(int8_t dir, int16_t speed) {
  speed = speed/100.0*255;
```

```
motor_L.run((9) == M1 ? -(dir*speed) : (dir*speed));
}

void mBot_setMotorRight(int8_t dir, int16_t speed) {
    speed = speed/100.0*255;
    motor_R.run((10) == M1 ? -(dir*speed) : (dir*speed));
}

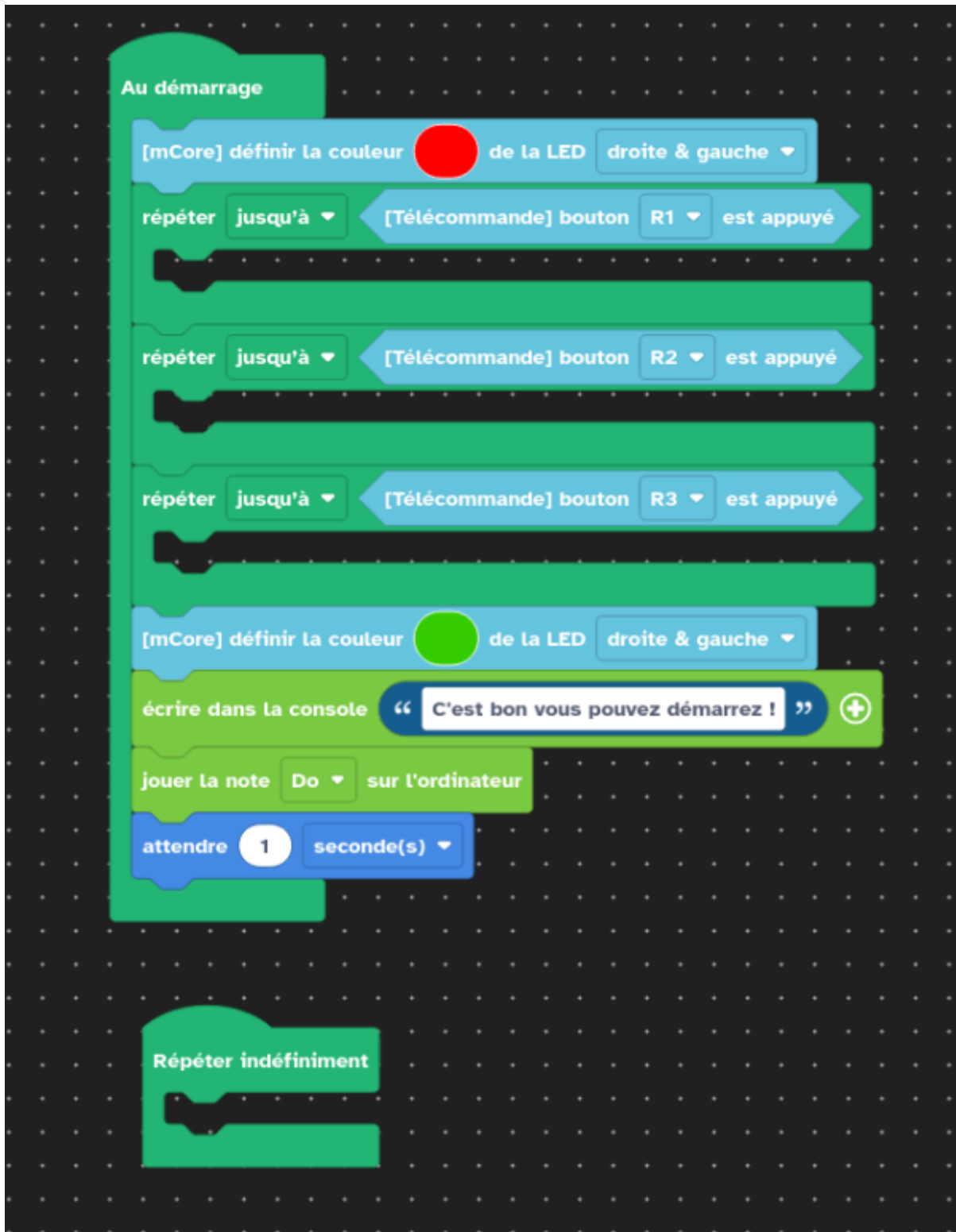
void setup() {
    ir.begin();
}

void loop() {
    if (ir.keyPressed(64)) {
        mBot_setMotorRight(1, 25);
        mBot_setMotorLeft(1, 25);
    } else if (ir.keyPressed(25)) {
        mBot_setMotorRight(-1, 25);
        mBot_setMotorLeft(-1, 25);
    } else if (ir.keyPressed(7)) {
        mBot_setMotorRight(1, 25);
        mBot_setMotorLeft(-1, 25);
    } else if (ir.keyPressed(9)) {
        mBot_setMotorRight(-1, 25);
        mBot_setMotorLeft(1, 25);
    } else {
        mBot_setMotorRight(0, 0);
        mBot_setMotorLeft(0, 0);
    }
}
```

## Mot de passe avec telecommande



Taper un mot de passe à 3 chiffres pour pouvoir utiliser votre robot



[mdprobor.ino](#)

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

MeRGBLed rgbled_board(7, 2);
```

```
MeIR ir;

void serial_setupConnection(long baudrate) {
  Serial.begin(baudrate);
  while (!Serial) {
    Serial.println("En attente de l'ouverture du port série...");
    delay(1000);
  }
  Serial.println("Port série activé. Baudrate: " + String(baudrate));
  delay(50);
}

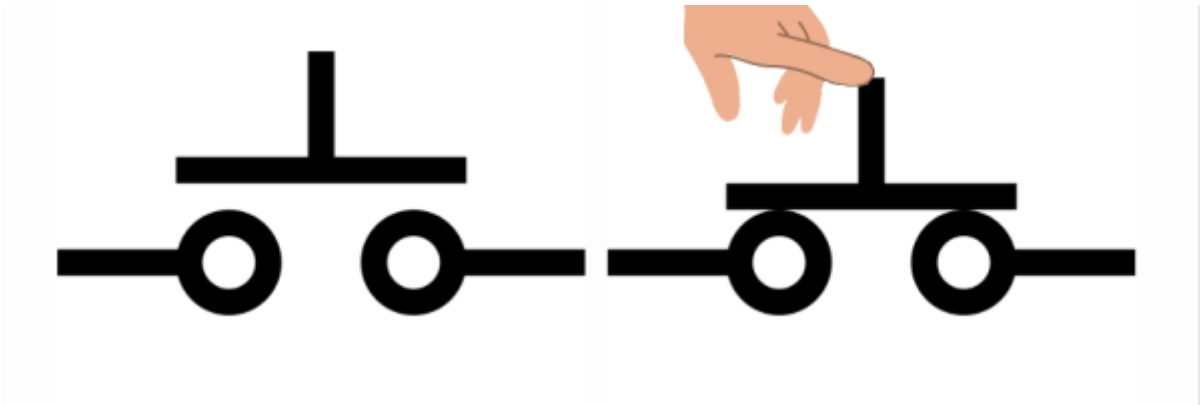
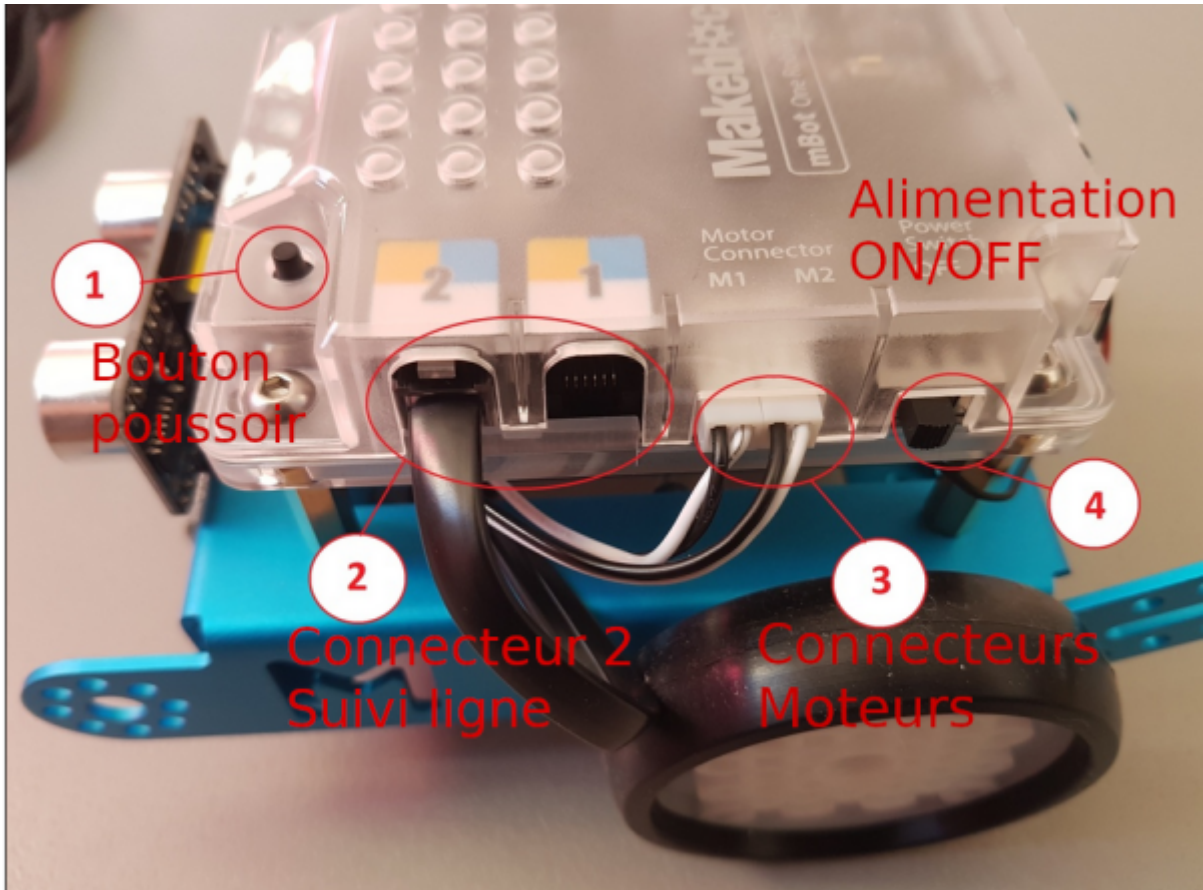
void setup() {
  ir.begin();
  serial_setupConnection(9600);
  rgbled_board.setColor(0, 255, 0, 0);
  rgbled_board.show();
  while (!(ir.keyPressed(12))) {
  }
  while (!(ir.keyPressed(24))) {
  }
  while (!(ir.keyPressed(94))) {
  }
  rgbled_board.setColor(0, 51, 204, 0);
  rgbled_board.show();
  Serial.println("C'est bon vous pouvez démarrez !");
  Serial.println("@music:261.63|");
  delay(1000*1);
}

void loop() {
}
```

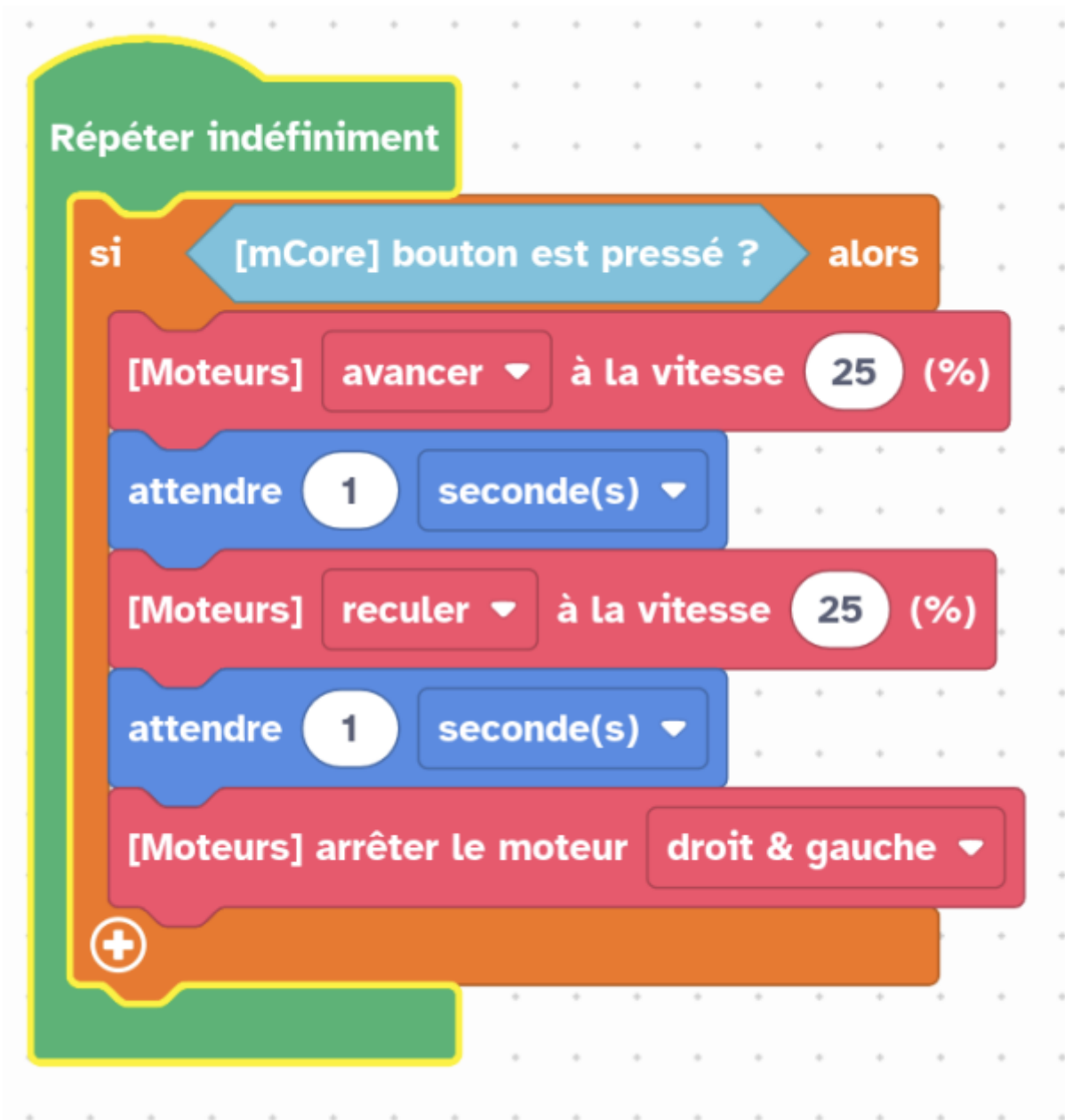
## robot-mbot-utiliser-le-bouton

[robot-mbot-utiliser-le-bouton](#)

robot\_mbot\_-\_utiliser\_le\_bouton.pdf



robot\_mbot\_-\_utiliser\_le\_bouton\_-\_partie\_3\_20251027\_15143.ino.tar  
(enlever .tar)



[boutonpoussoir.ino](#)

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

MeDCMotor motor_L(9);
MeDCMotor motor_R(10);

int buttonPressed() {
  return analogRead(A7) <= 10 ? 1 : 0;
}

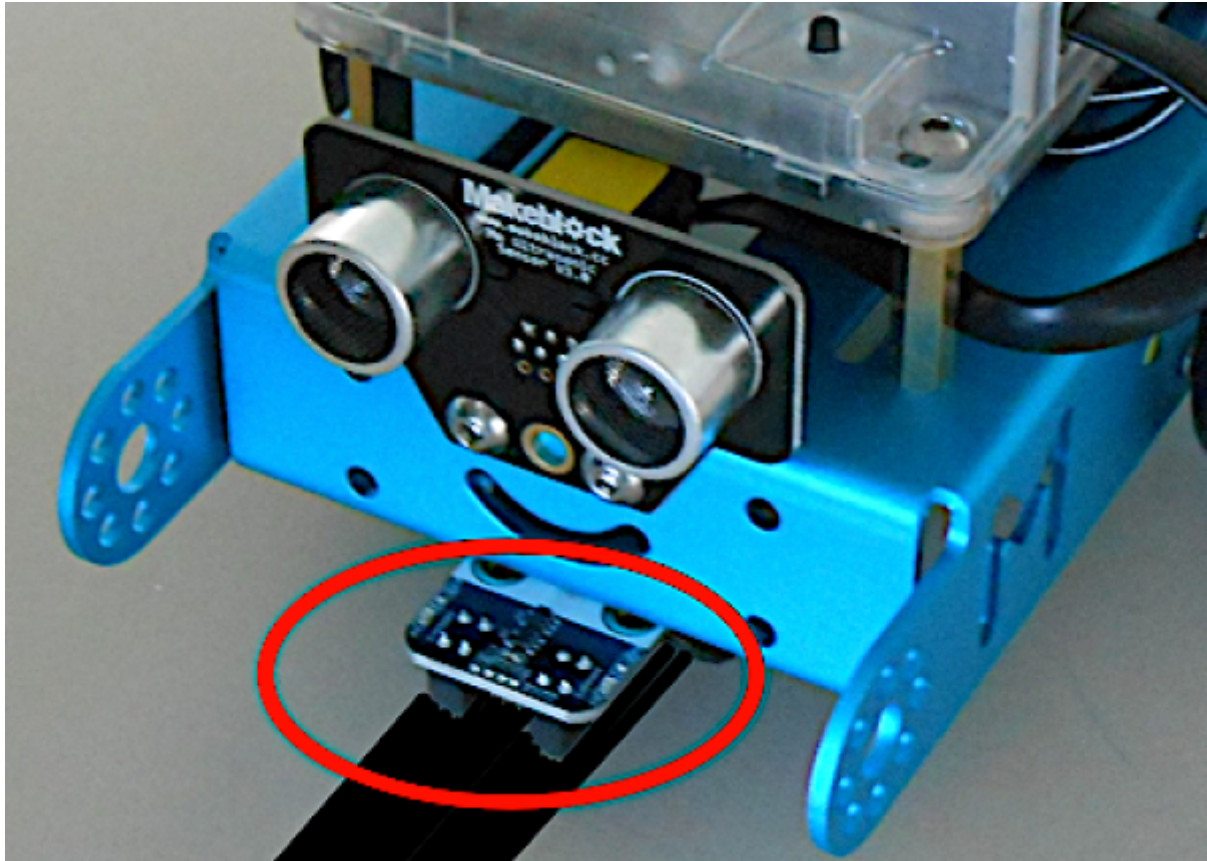
void mBot_setMotorLeft(int8_t dir, int16_t speed) {
  speed = speed/100.0*255;
  motor_L.run((9) == M1 ? -(dir*speed) : (dir*speed));
}
```

```
}  
  
void mBot_setMotorRight(int8_t dir, int16_t speed) {  
    speed = speed/100.0*255;  
    motor_R.run((10) == M1 ? -(dir*speed) : (dir*speed));  
}  
  
void setup() {  
    pinMode(A7, INPUT);  
}  
  
void loop() {  
    if (buttonPressed()) {  
        mBot_setMotorRight(1, 25);  
        mBot_setMotorLeft(1, 25);  
        delay(1000*1);  
        mBot_setMotorRight(-1, 25);  
        mBot_setMotorLeft(-1, 25);  
        delay(1000*1);  
        mBot_setMotorRight(0, 0);  
        mBot_setMotorLeft(0, 0);  
    }  
}
```

## Robot suivi Ligne

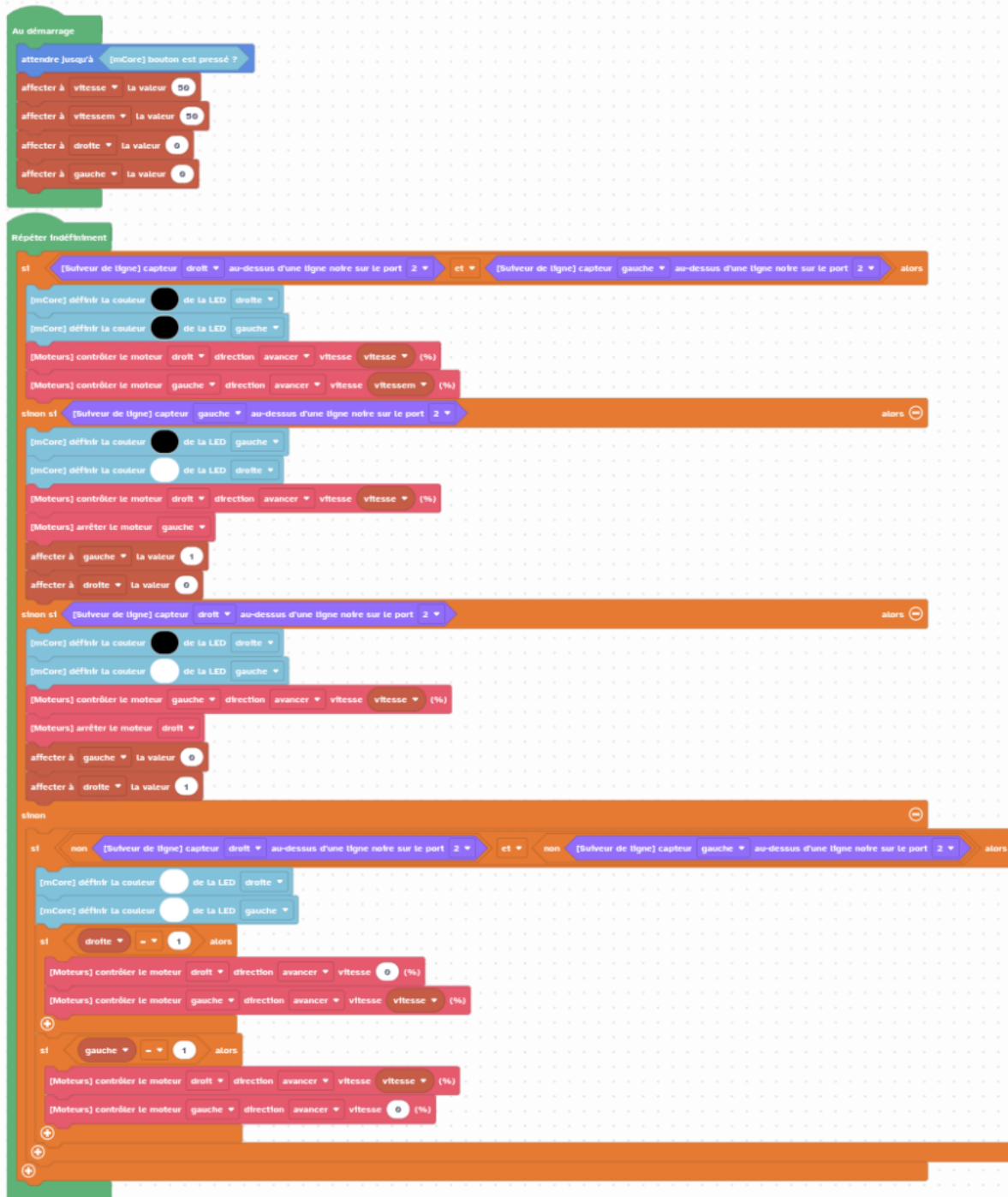
[robot-mbot-utiliser-les-suiveurs-de-ligne](#)

robot\_mbot\_-\_utiliser\_les\_suiveurs\_de\_ligne.pdf



correctionsuiviligneexo4\_vittascience\_20251020\_193057.ino.zip

## Robot circuit Vittascience test gauchedroite variables 20251121



[suivilignevariable.ino](#)

```
#include <MeMCore.h>
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

// Line Finder on PORT_2
```

```
MeLineFollower lineFinder_2(PORT_2);
MeRGBLed rgbled_board(7, 2);
MeDCMotor motor_L(9);
MeDCMotor motor_R(10);

int vitesse;
int vitessesm;
int droite;
int gauche;

int buttonPressed() {
    return analogRead(A7) <= 10 ? 1 : 0;
}

void mBot_setMotorLeft(int8_t dir, int16_t speed) {
    speed = speed/100.0*255;
    motor_L.run((9) == M1 ? -(dir*speed) : (dir*speed));
}

void mBot_setMotorRight(int8_t dir, int16_t speed) {
    speed = speed/100.0*255;
    motor_R.run((10) == M1 ? -(dir*speed) : (dir*speed));
}

void setup() {
    pinMode(A7, INPUT);
    while (!buttonPressed() ) {}
    vitesse = 50;
    vitessesm = 50;
    droite = 0;
    gauche = 0;
}

void loop() {
    if (!lineFinder_2.readSensor2() && !lineFinder_2.readSensor1()) {
        rgbled_board.setColor(1, 0, 0, 0);
        rgbled_board.show();
        rgbled_board.setColor(2, 0, 0, 0);
        rgbled_board.show();
        mBot_setMotorRight(1, vitesse);
        mBot_setMotorLeft(1, vitessesm);
    } else if (!lineFinder_2.readSensor1()) {
        rgbled_board.setColor(2, 0, 0, 0);
        rgbled_board.show();
        rgbled_board.setColor(1, 255, 255, 255);
        rgbled_board.show();
        mBot_setMotorRight(1, vitesse);
        mBot_setMotorLeft(0, 0);
        gauche = 1;
        droite = 0;
    }
}
```

```
} else if (!!lineFinder_2.readSensor2()) {
  rgbled_board.setColor(1, 0, 0, 0);
  rgbled_board.show();
  rgbled_board.setColor(2, 255, 255, 255);
  rgbled_board.show();
  mBot_setMotorLeft(1, vitesse);
  mBot_setMotorRight(0, 0);
  gauche = 0;
  droite = 1;
} else {
  if (!!lineFinder_2.readSensor2() && !!lineFinder_2.readSensor1()) {
    rgbled_board.setColor(1, 255, 255, 255);
    rgbled_board.show();
    rgbled_board.setColor(2, 255, 255, 255);
    rgbled_board.show();
    if (droite == 1) {
      mBot_setMotorRight(1, 0);
      mBot_setMotorLeft(1, vitesse);
    }
    if (gauche == 1) {
      mBot_setMotorRight(1, vitesse);
      mBot_setMotorLeft(1, 0);
    }
  }
}
```

## Sortie d un Labyrinthe avec un mBot

[Sortie d un Labyrinthe avec un mBot](#)

From: <https://chanterie37.fr/fablab37110/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link: <https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:cours:vittascience&rev=1772965517>

Last update: 2026/03/08 11:25

