Vittascience

Vittascience FR

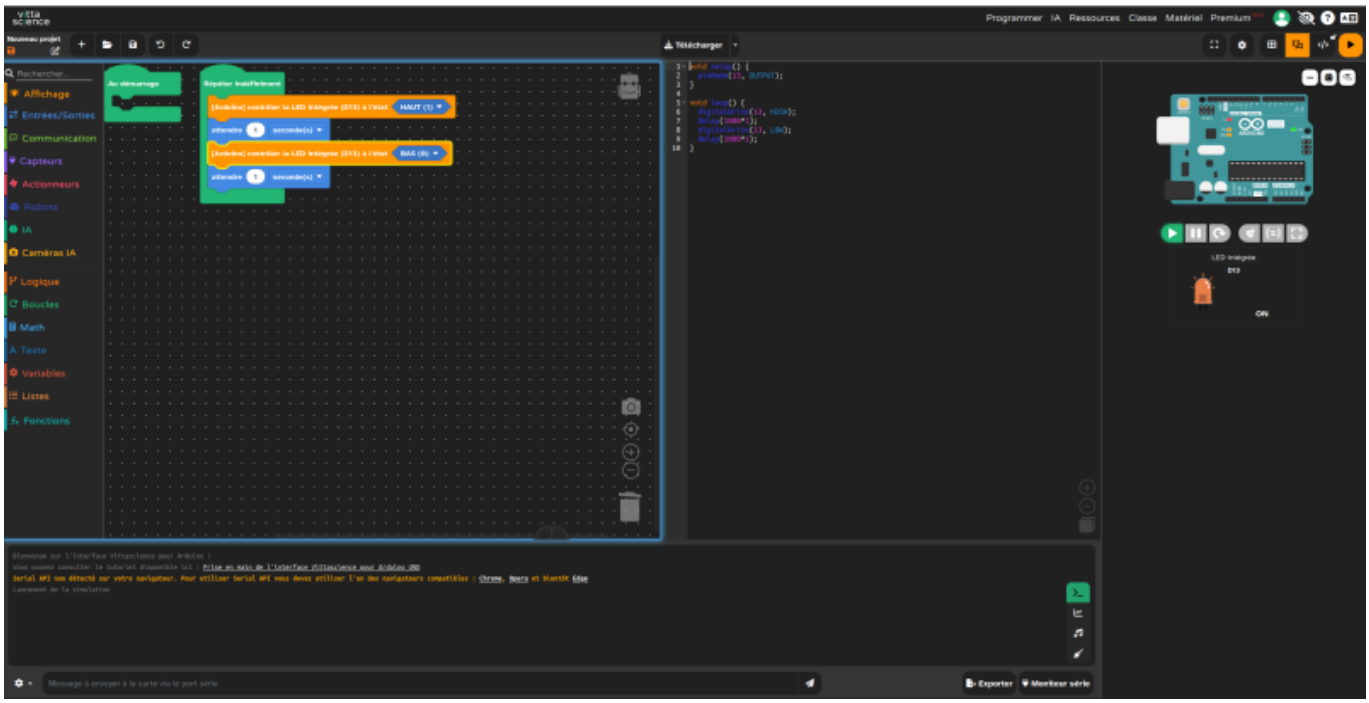
Vittascience : Une plateforme innovante pour enseigner la programmation

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

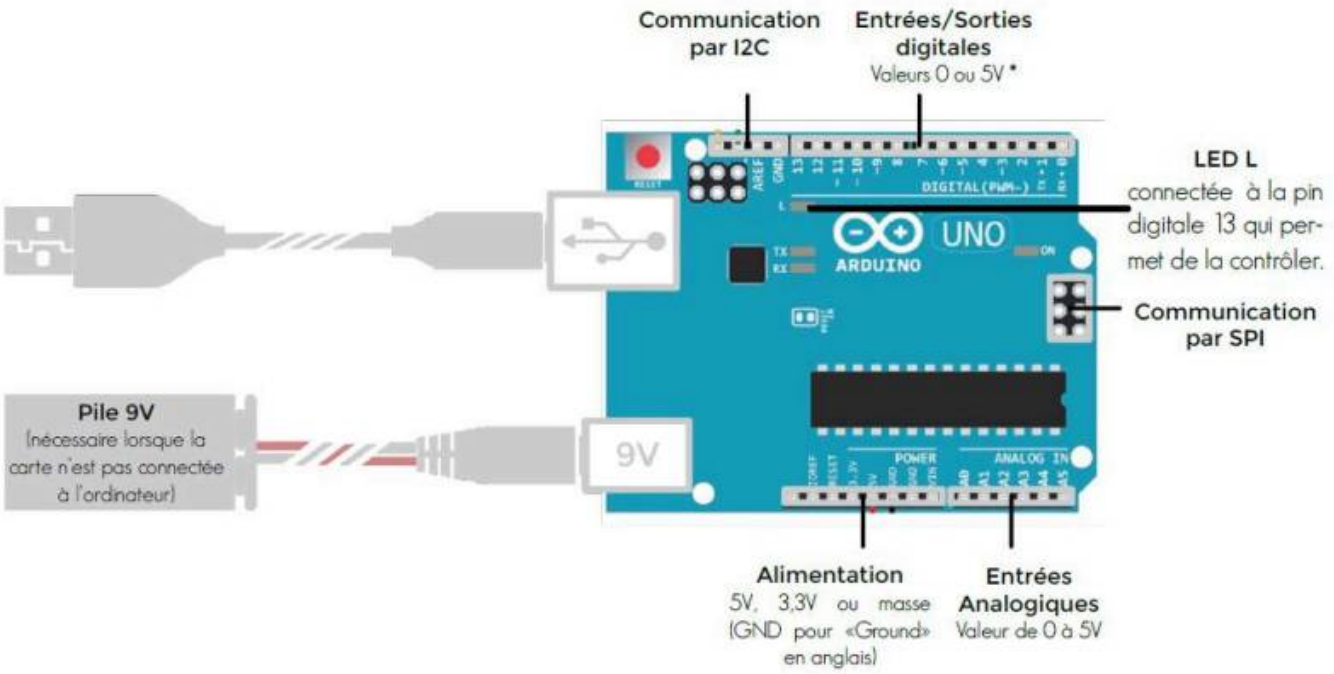
Programmer un arduino

Vittascience et Arduino



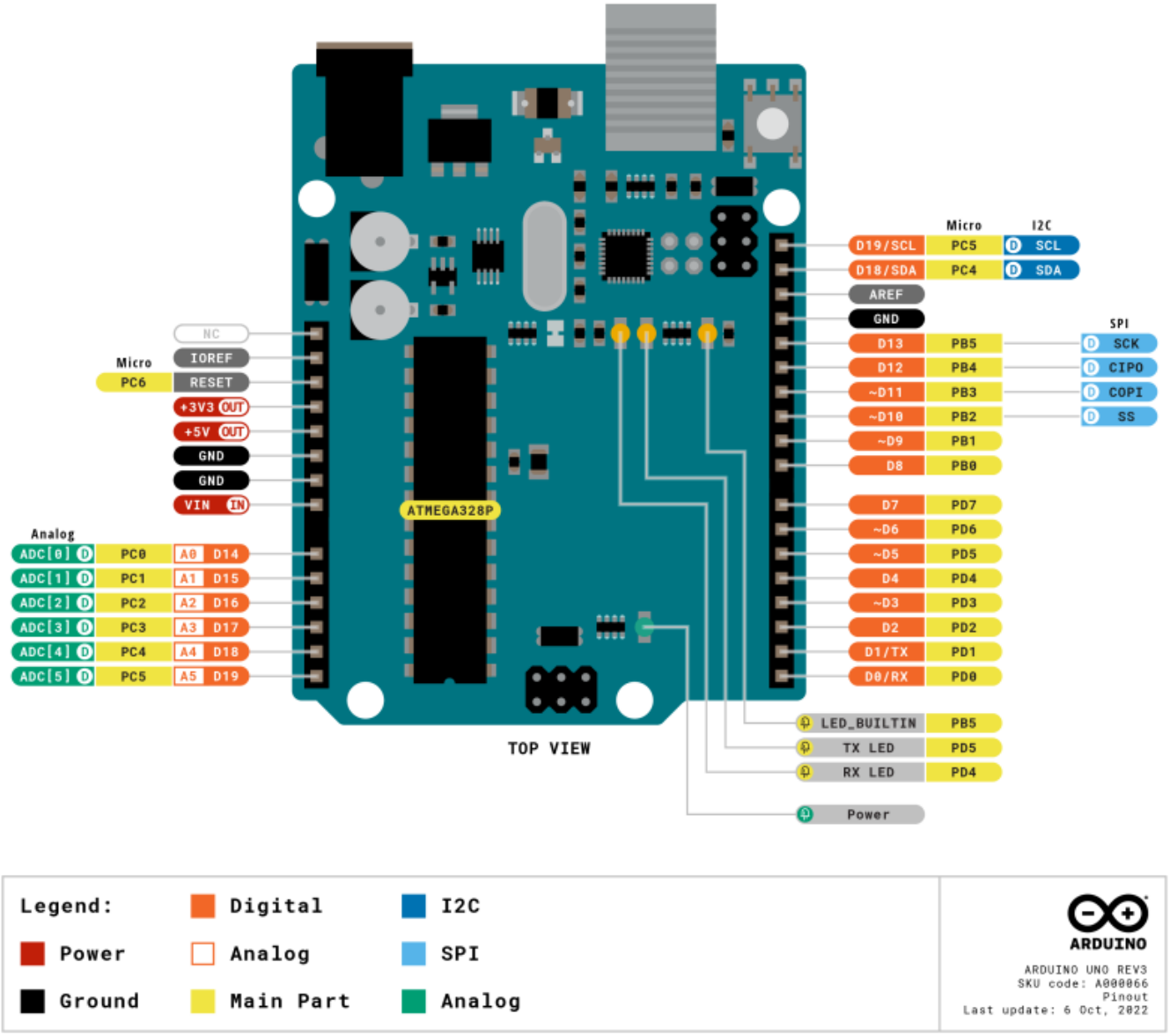
Decouverte Arduino

Manipulation de la carte Arduino Uno et de sa LED intégrée



Decouverte Arduino UNO

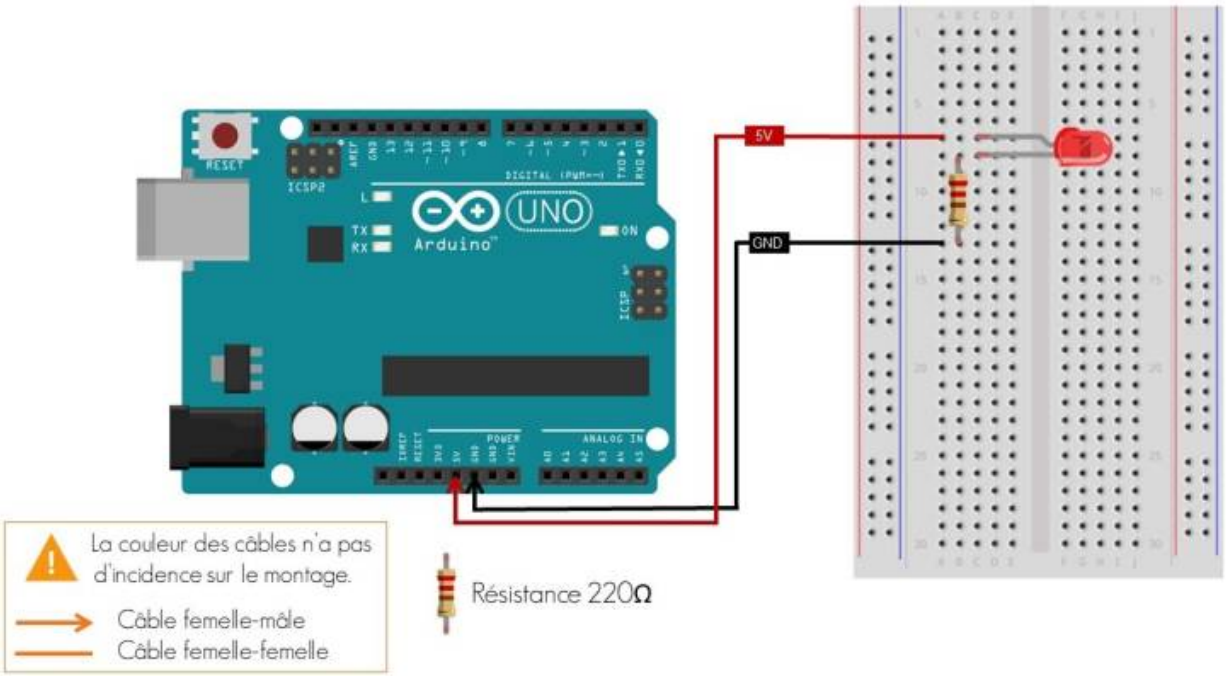
Prise en main de la carte Arduino UNO et de l'interface Vittascience



prise_en_main_de_la_carte_arduino_uno_et_de_l_interface_vittascience.pdf

Decouverte LED et Arduino

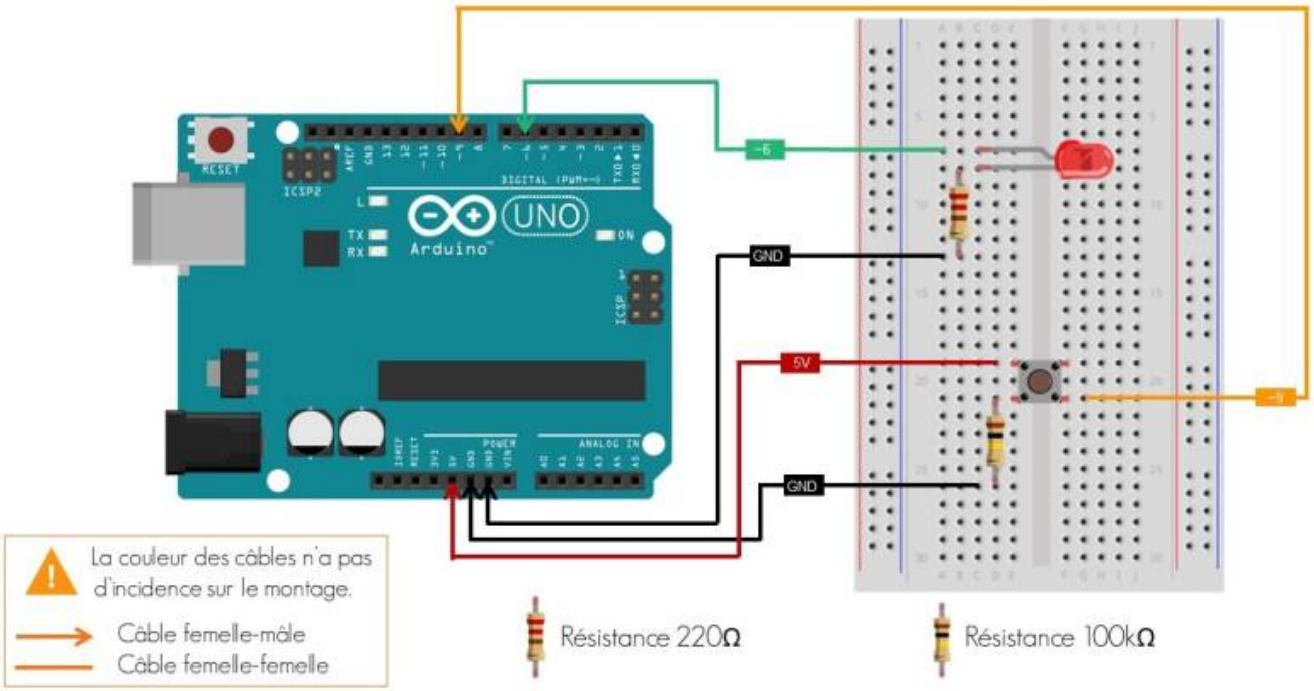
Allumer une LED avec Arduino



allumer_une_led_avec_arduino.pdf

Decouverte Bouton + LED + Arduino

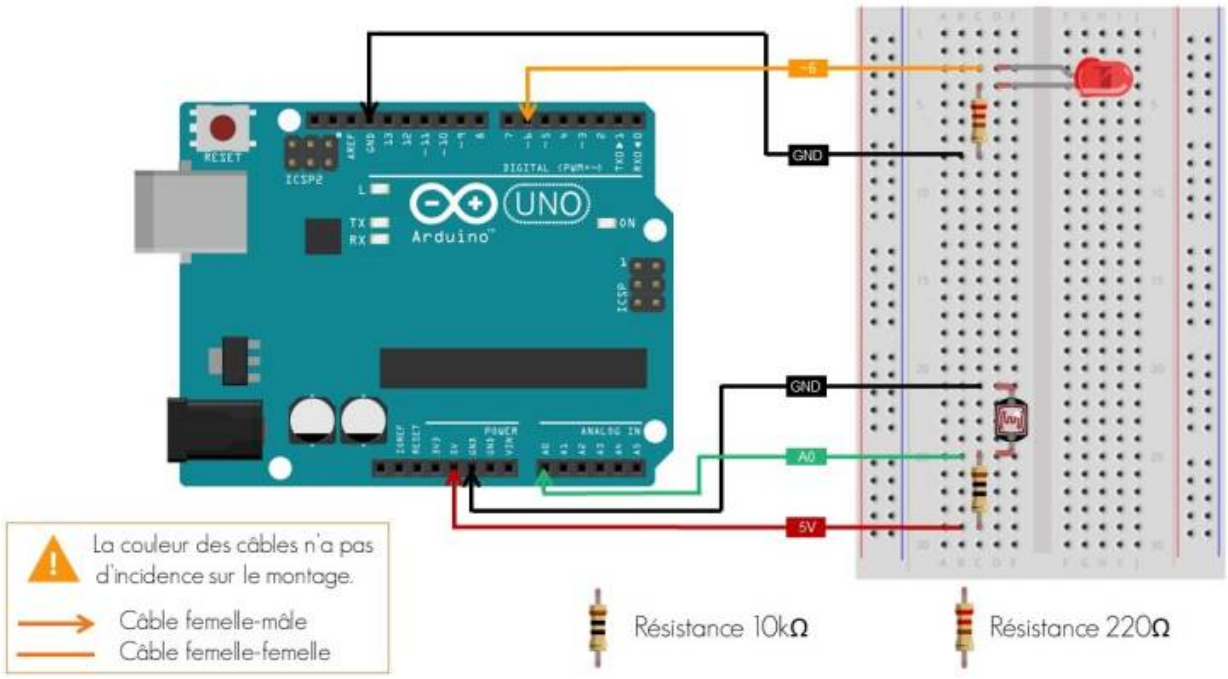
Ajouter un bouton sur Arduino



ajouter_un_bouton_sur_arduino.pdf

Decouverte PhotoResistance + Led + Arduino

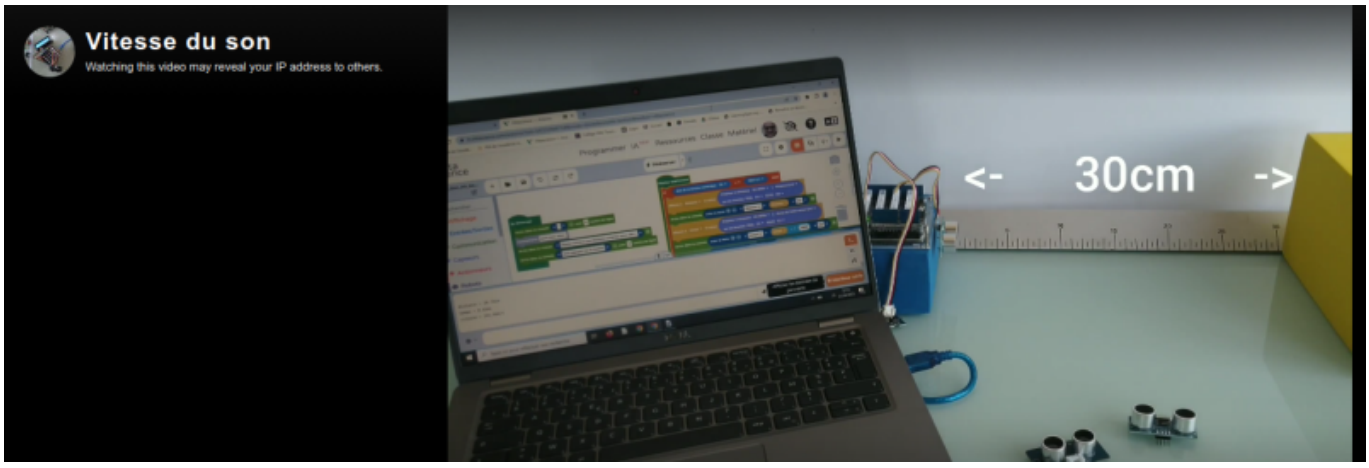
Mesurer la luminosité avec une photorésistance sur Arduino



mesurer_la_luminosite_avec_une_photoresistance_sur_arduino.pdf

Decouverte Vitesse du son avec un HC-SR04 + Arduino

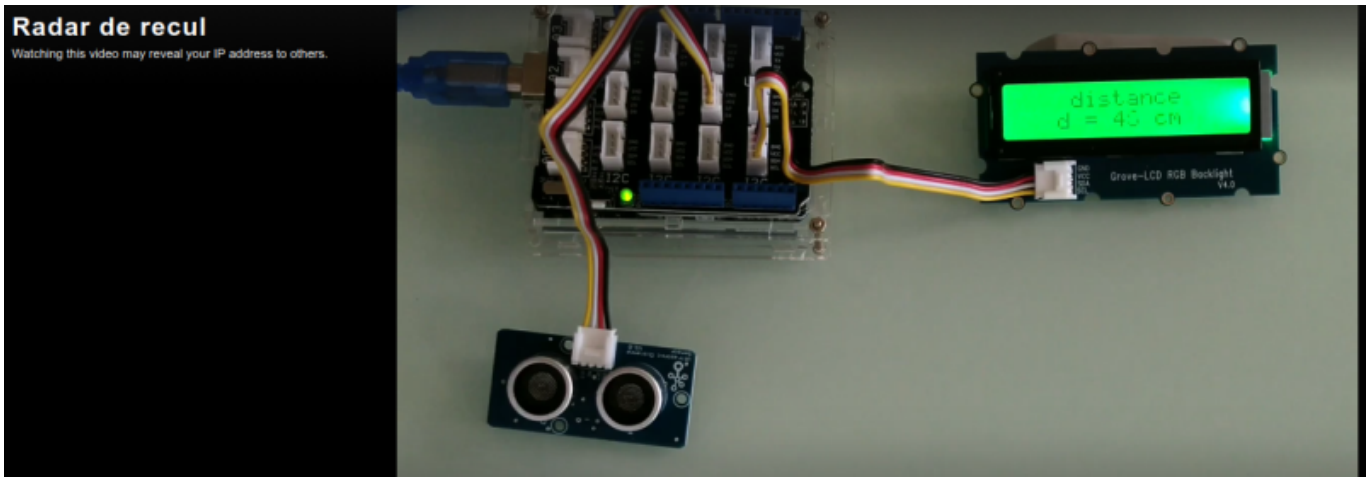
Vitesse du son : Capteur ultrason HC-SR04



vitesse_du_son.pdf

Decouverte Radar de recul avec HC-SR04 + Arduino

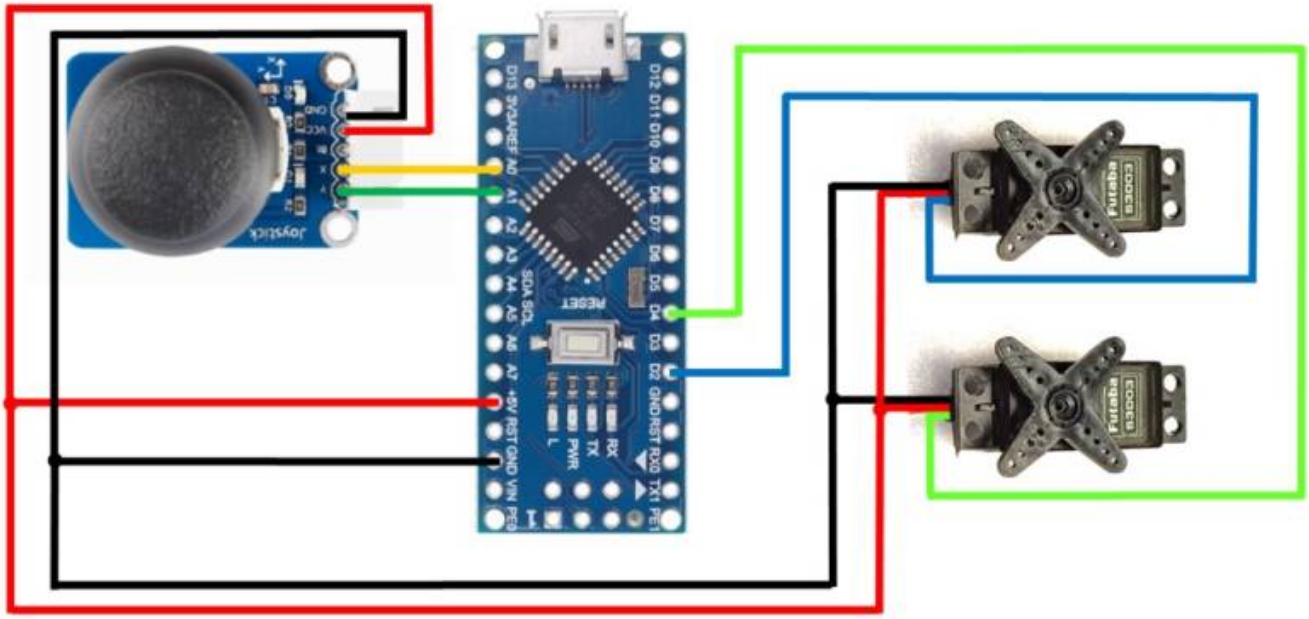
Radar de recul avec un écran LCD



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

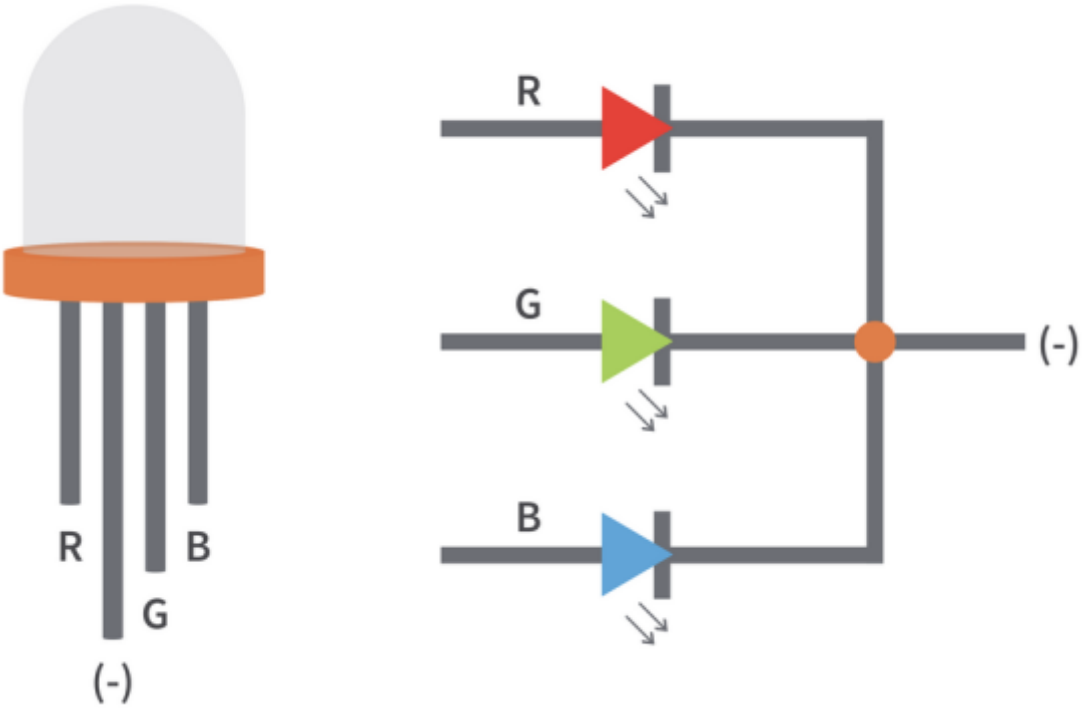
Image fond ecran pour virtualisation mBot1 sur Vittascience

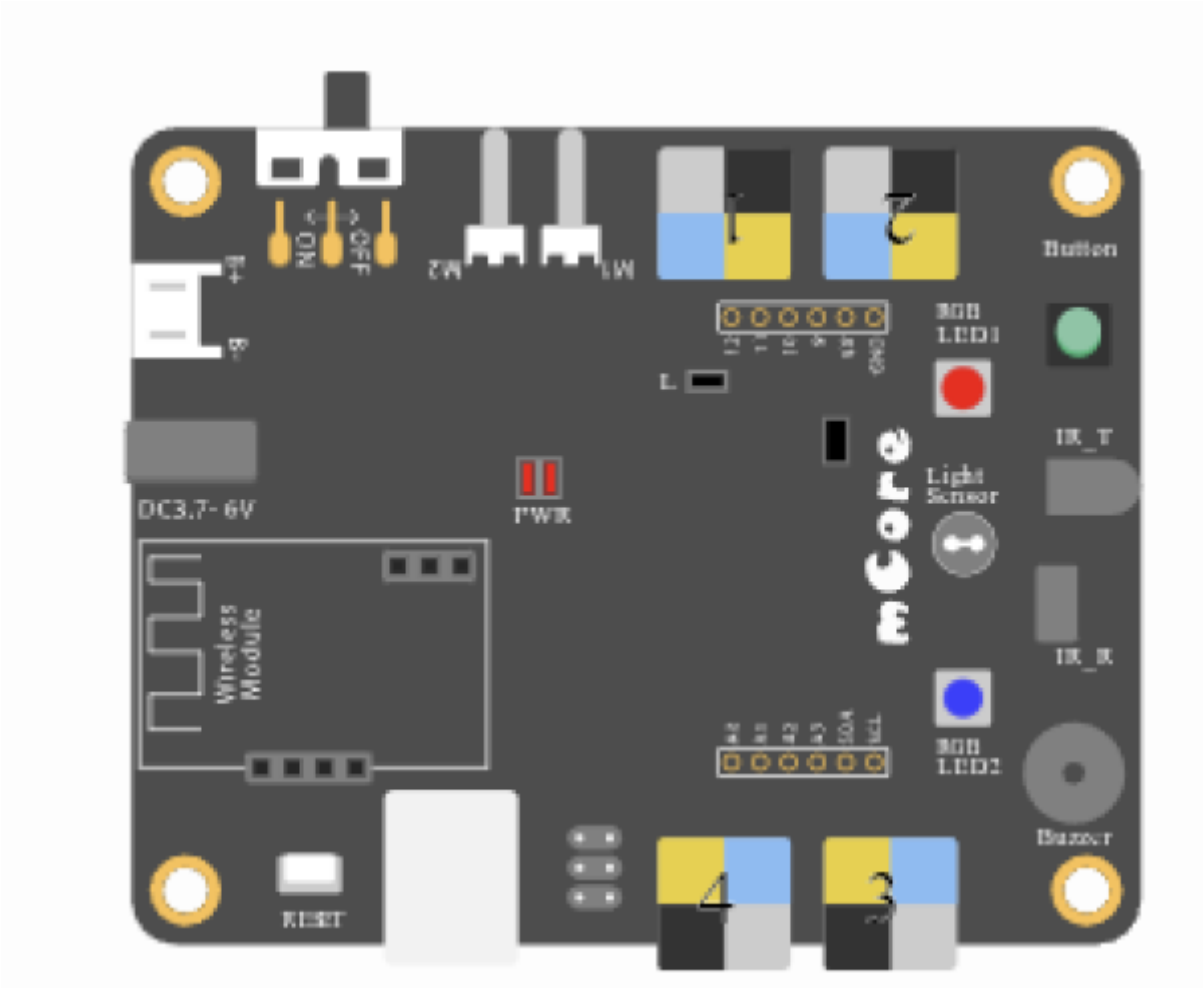
tracerobot004.jpg.zip

Programmes mBot1 Vittascience

robot-mbot-utiliser-les-led-rgb

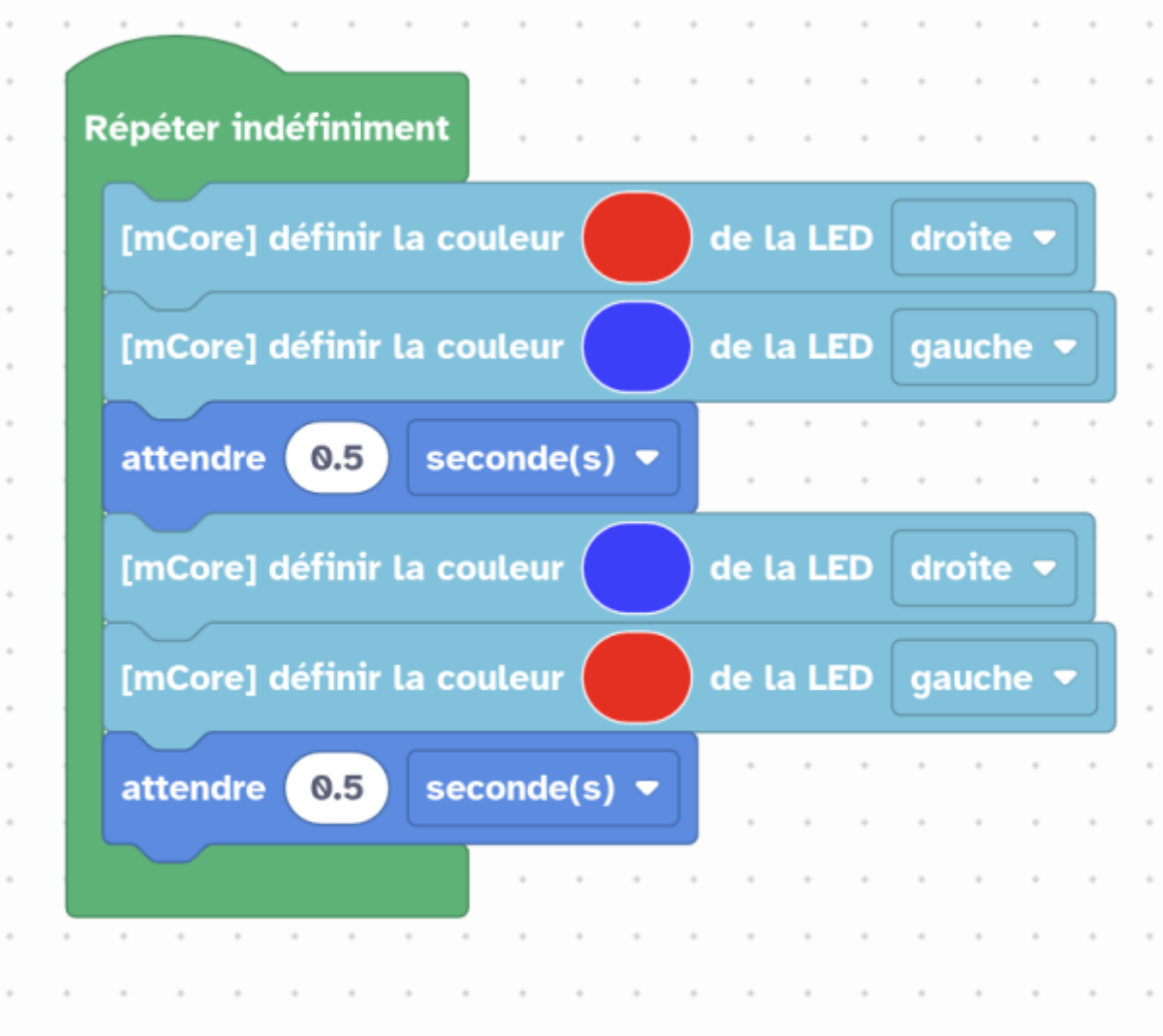






robot-mbot-utiliser-les-led-rgb

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);
}
```

Exercice- 1 : Le feu tricolore RGB du mBot

Objectif : programmer les LED RGB du mBot pour afficher plusieurs couleurs, puis les éteindre.

Matériel : un mBot classique avec sa carte mCore et un ordinateur équipé de Chrome pour se connecter à Vittascience. La carte mCore possède deux LED RGB contrôlables séparément ou ensemble. Les couleurs se règlent avec des valeurs de rouge, vert et bleu, de 0 à 255. WordPress

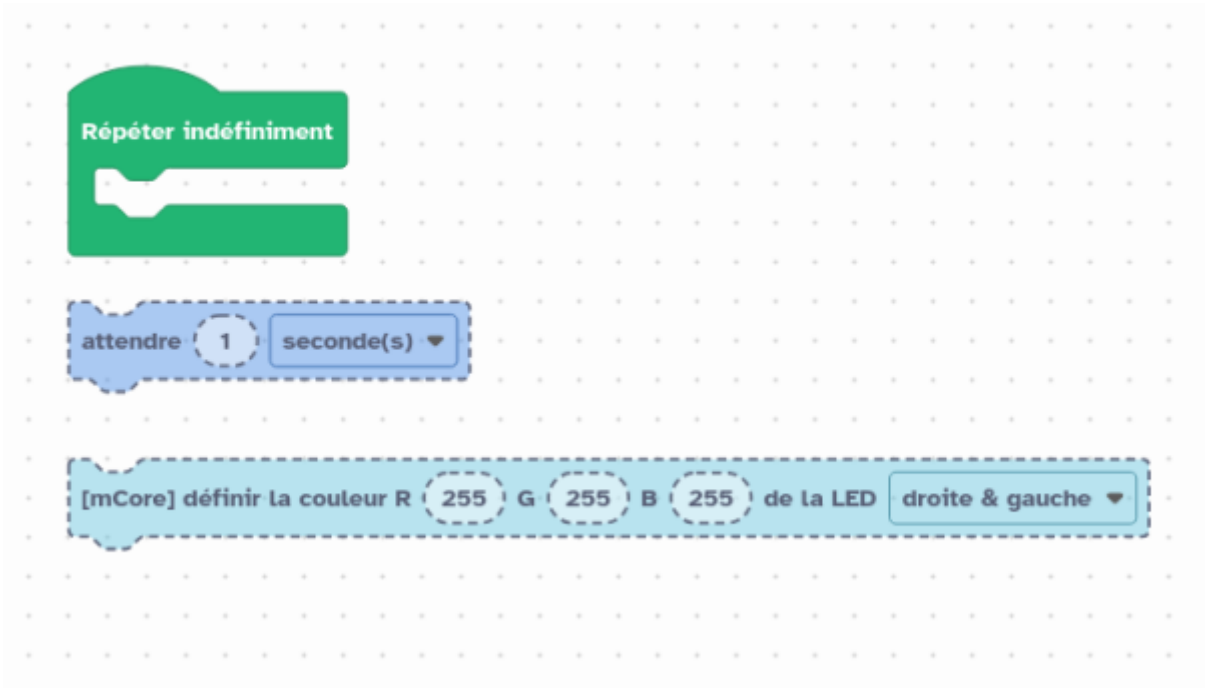
Consigne : crée dans mBlock un programme qui affiche chaque couleur pendant une seconde, puis éteint les LED.

- Choisis l'événement de démarrage du programme.
- Ajoute un bloc pour régler les LED RGB sur rouge : rouge 255, vert 0, bleu 0.
- Ajoute une attente d'1 seconde.
- Recommence avec :
  - vert : 0, 255, 0 ;
  - bleu : 0, 0, 255 ;
  - jaune : 255, 255, 0.
- Pour finir, règle les trois valeurs à 0 afin d'éteindre les LED.

Pour continuer :

Ajoute une boucle pour répéter le feu tricolore **trois fois**, et ensuite choisis deux couleurs et alterne-les

Utilise les blocs suivant :





Robot mBot : utiliser le buzzer

Correspondance des notes avec la frequence en Hz

À 440 hertz :

Notes	Fréquences
DO (C4)	261.63 Hz
RÉ (D4)	293.66 Hz
MI (E4)	329.63 Hz
FA (F4)	349.23 Hz
SOL (G4)	392 Hz
LA (A4)	440 Hz
SI (B4)	493.88 Hz
Do aigu	523,25 Hz

Exercice- 2 : Jouer de la musique avec le buzzer du mBot

Objectif : programmer le Buzzer du mBot pour jouer une musique.

Matériel : un mBot classique avec sa carte mCore et un ordinateur équipé de Chrome pour se connecter à Vittascience. La carte mCore possède un Buzzer pour émettre un son très basique

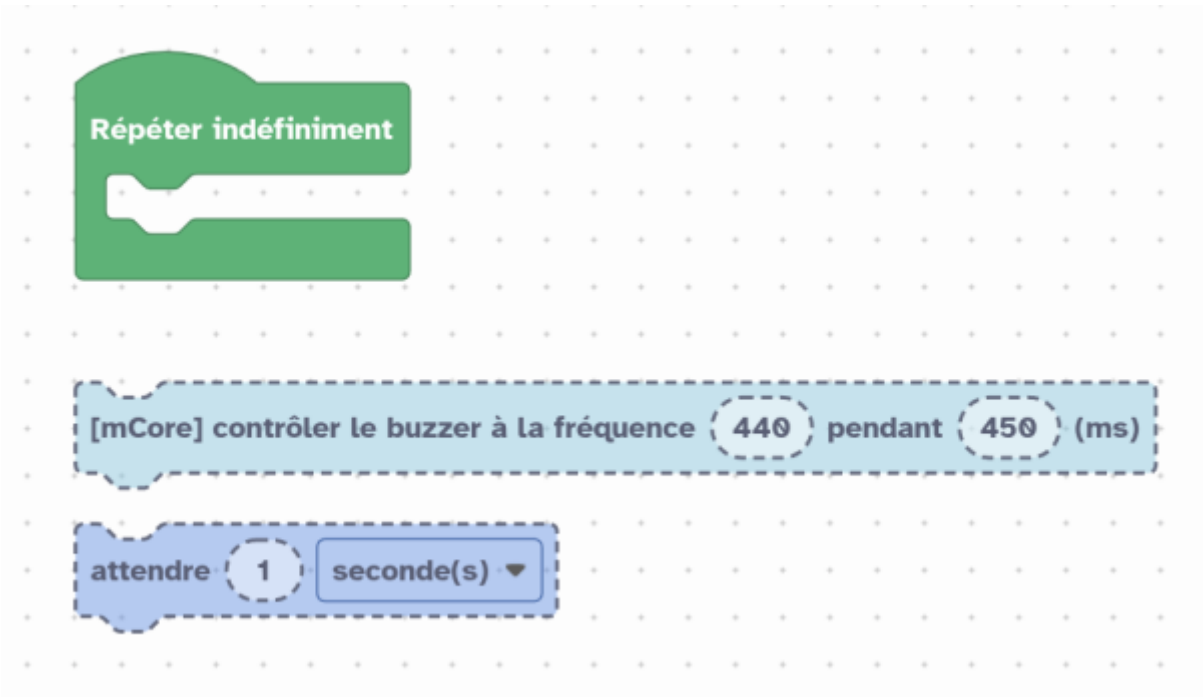
Consigne : crée dans mBlock un programme qui joue une petite musique .

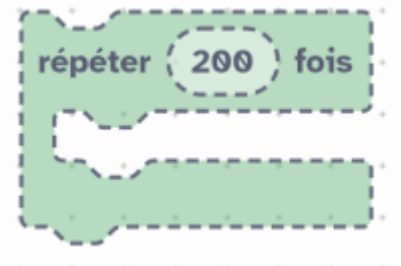
- Choisis l'événement de démarrage du programme.
- Ajoute un bloc pour régler le buzzer " Contrôler le buzzer à la fréquence XXX pendant xxxx ms".
- Ajoute une attente en fonction du résultat voulu
- un petit morceau très court : « Frère Jacques », début de la mélodie.
  - Do — 261,63 Hz — 400 ms
  - Ré — 293,66 Hz — 400 ms
  - Mi — 329,63 Hz — 400 ms
  - Do — 261,63 Hz — 400 ms

Pour continuer :

Ajoute une boucle pour répéter la musique **cinq fois**.

Utilise les blocs suivant :



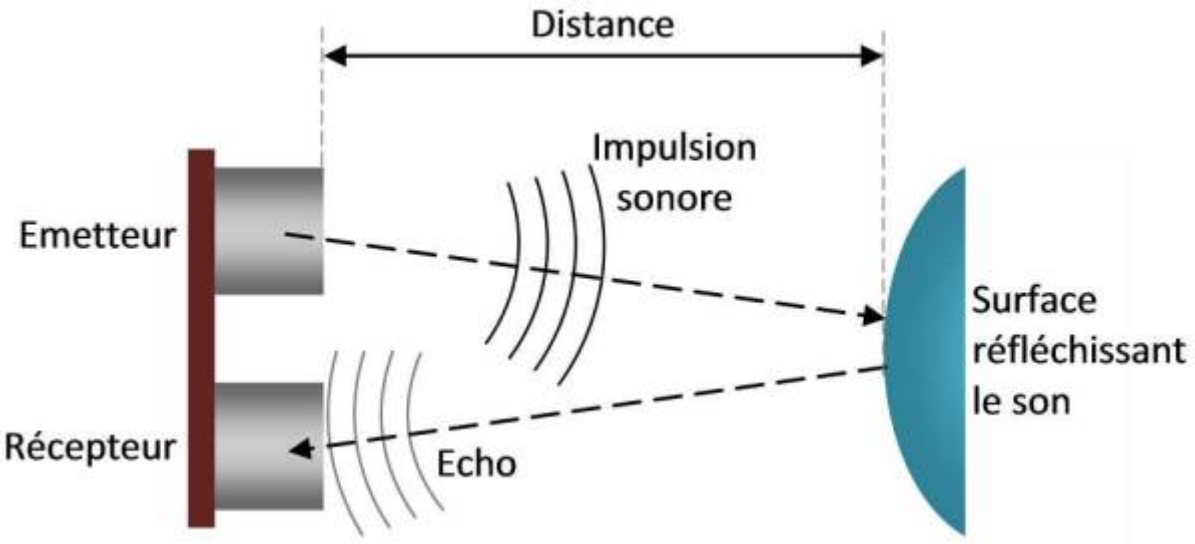


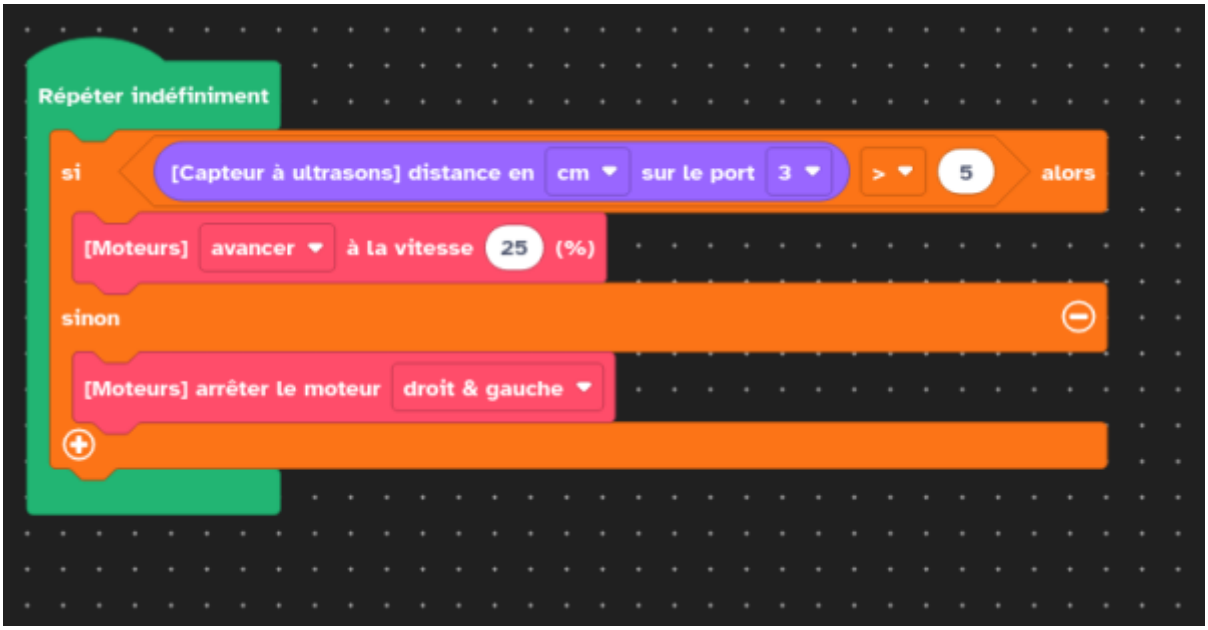
Robot mBot - Utiliser le capteur de distance

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);
  }
}
```

Exercice- 3 : Mesurer une distance avec le robot mBot

Objectif : programmer le capteur de distance du mBot pour afficher le résultat dans la console.

Matériel : un mBot classique avec sa carte mCore et un ordinateur équipé de Chrome pour se connecter à Vittascience. La carte mCore possède un capteur de distance HC-SR04

Consigne : crée dans Vittascience un programme qui affiche la distance mesurée .

- Choisis l'événement de démarrage du programme.
- Ajoute un bloc d'affichage/console
- Ouvre le moniteur série en bas à droite pour voir les mesures.
- Met un obstacle devant le capteur de distance
- Vérifie la distance en la mesurant avec un double decimetre (pour vérifier l'erreur ...)

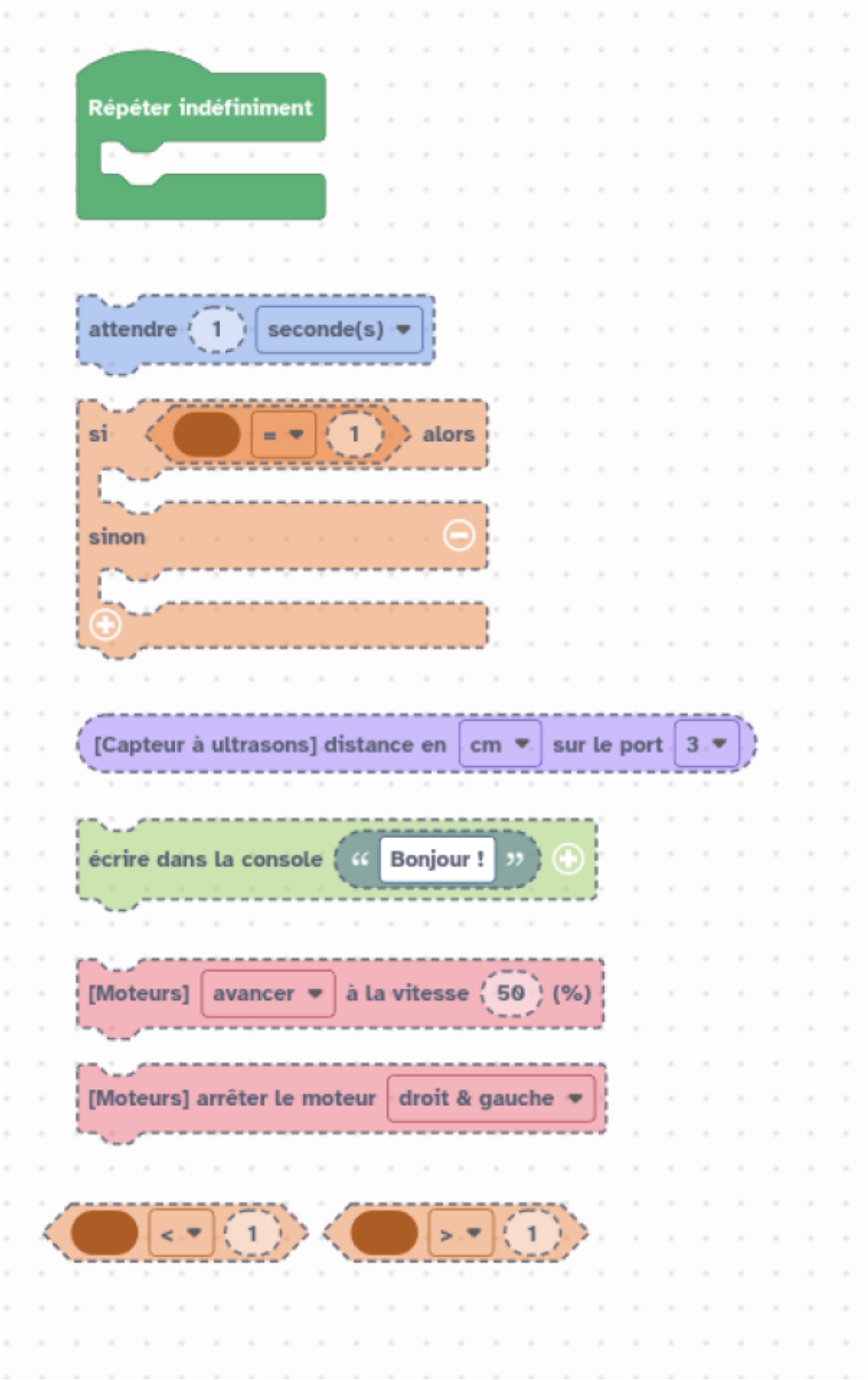
Pour continuer :

Créer un programme anti-obstacle qui arrête le mBot si la distance est < à 5cm est qui fait avancer le mBot si la distance est > à 5cm

algorithmedistance001.txt

```
Au démarrage
Répéter indéfiniment
    Si la distance du capteur ultrason (port 3) est inférieure à 5 cm
        Arrêter les moteurs
    Sinon
        Faire avancer le mBot à 50 % de puissance
```

Utiliser les blocs suivant


Répéter indéfiniment
attendre 1 seconde(s)
si = 1 alors
sinon
[Capteur à ultrasons] distance en cm sur le port 3
écrire dans la console “ Bonjour ! ”
[Moteurs] avancer à la vitesse 50 (%)
[Moteurs] arrêter le moteur droit & gauche
< 1
> 1

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



Exercice- 4 : Tester la telecommande du robot mBot

Objectif : programmer la telecommande du mBot pour verifier son fonctionnement.

Matériel : un mBot classique avec sa carte mCore et un ordinateur équipé de Chrome pour se connecter à Vittascience. La carte mCore possède un capteur de infrarouge pour la liaison avec une telecommande

Consigne : crée dans Vittascience un programme qui test la telecommande .

algorithmetelecommande001.txt

```
Au démarrage
Répéter indéfiniment
    Si le bouton A de la télécommande est appuyé
        Allumer les LED en rouge
    Sinon
        Éteindre les LED
```

Pour continuer :

Créer un programme de verrouillage par télécommande : le code est 1–2–3. Tant qu'il n'est pas saisi, le mBot reste arrêté.

algoriyhmetelecommande002.txt

```
Au démarrage
    mettre étape à 0
    mettre autorisé à faux
    arrêter les moteurs

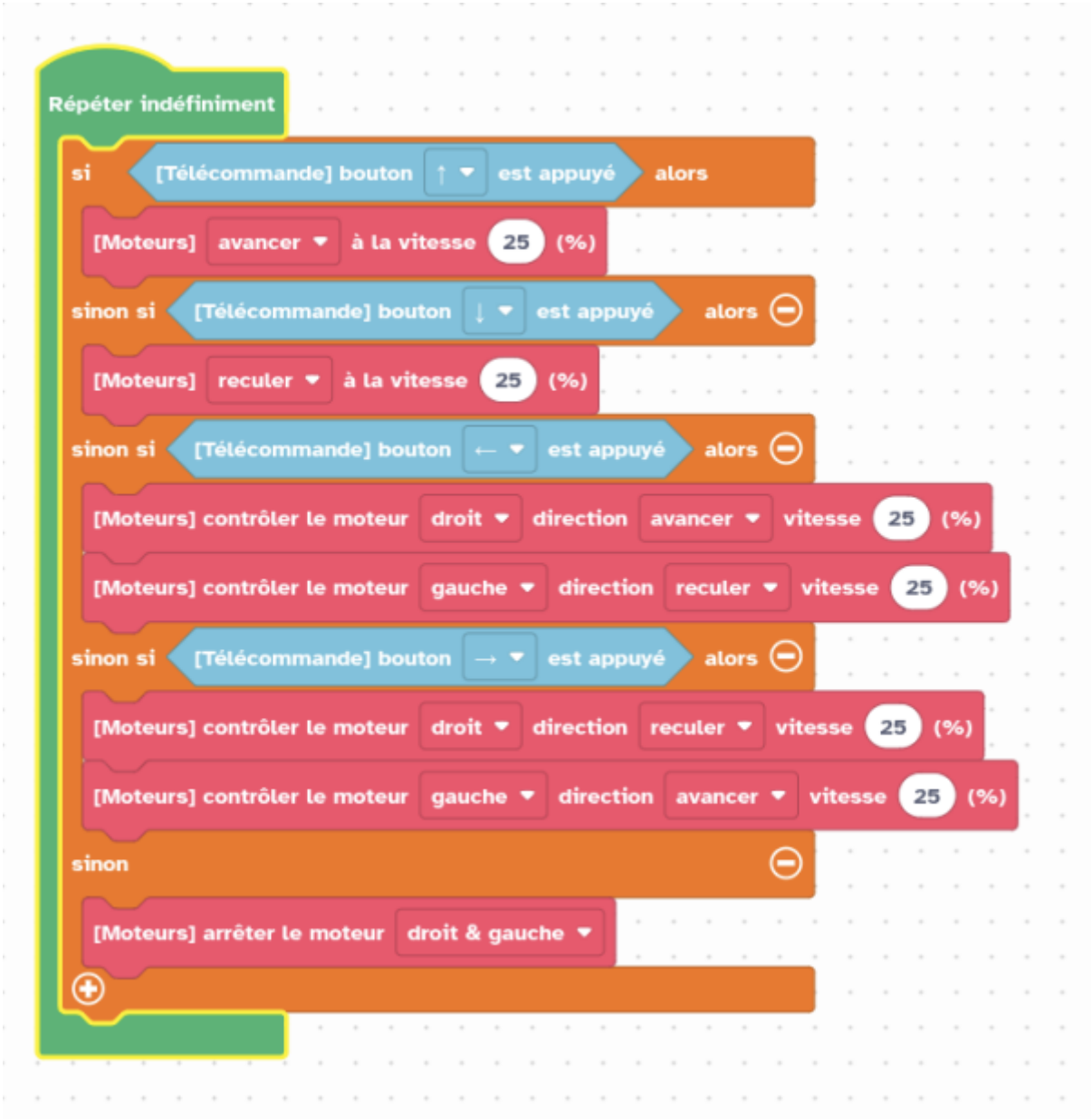
Répéter indéfiniment
    Si autorisé = faux :
        Si le bouton 1 est appuyé :
            si étape = 0, mettre étape à 1
            sinon, remettre étape à 0

        Si le bouton 2 est appuyé :
            si étape = 1, mettre étape à 2
            sinon, remettre étape à 0

        Si le bouton 3 est appuyé :
            si étape = 2, mettre autorisé à vrai
            sinon, remettre étape à 0

    Si autorisé = vrai :
        piloter le mBot avec les flèches de la télécommande
```

Utiliser les blocs suivant



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));
}

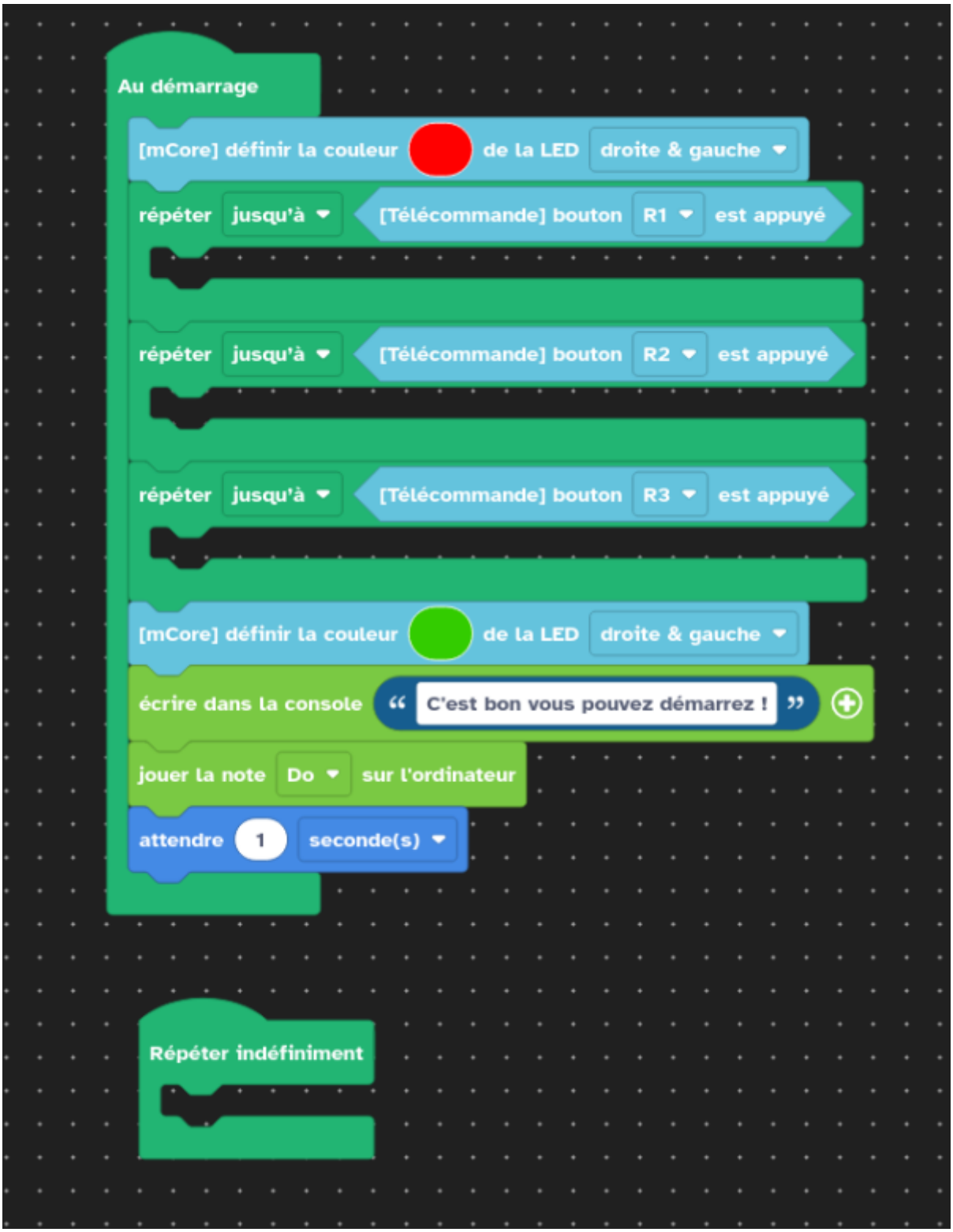

void setup() {
  ir.begin();
}

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



mdp_robot.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);
}

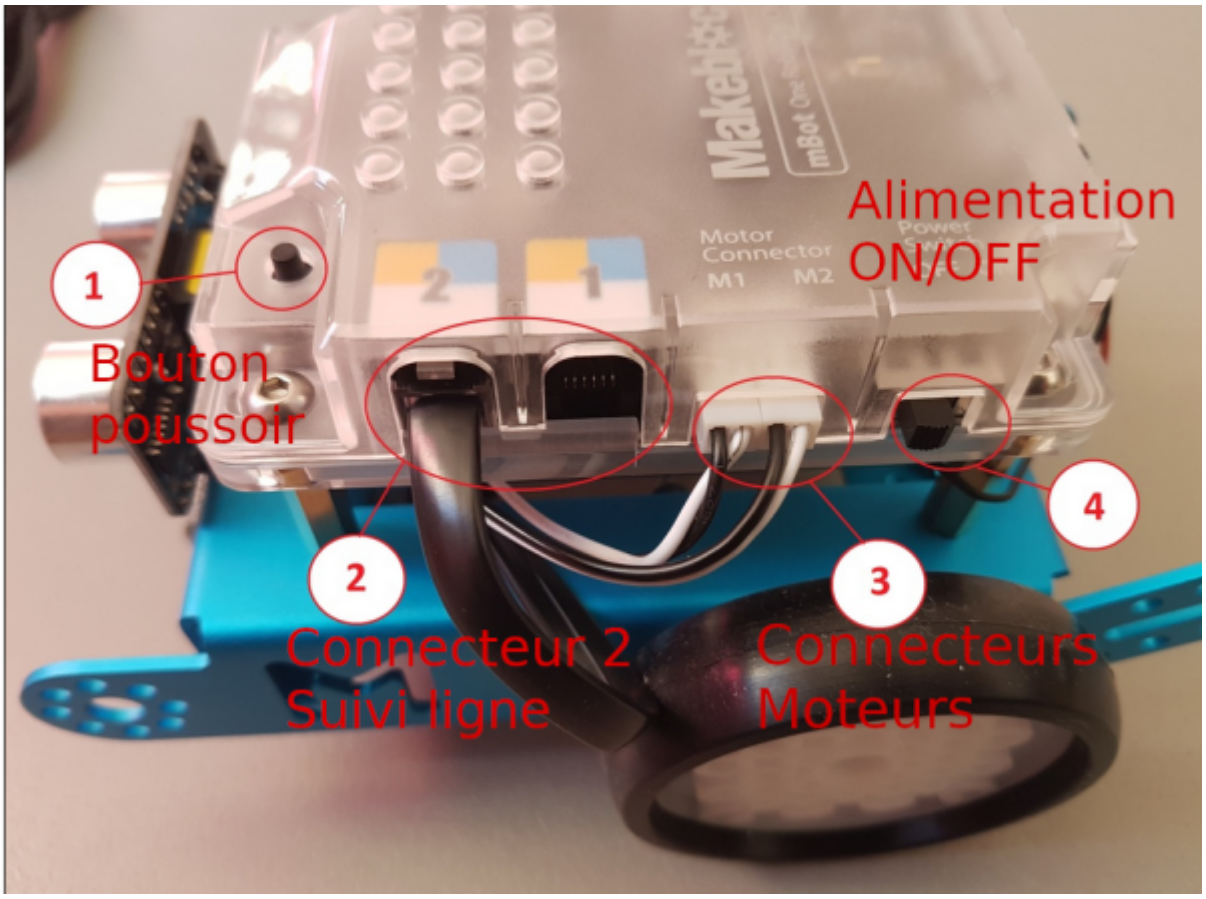

void setup() {
  ir.begin();
  serial_setupConnection(9600);
  rgbled_board.setColor(0, 255, 0, 0);
  rgbled_board.show();
  while (!(ir.keyPressed(12))) {
  }
  while (!(ir.keyPressed(24))) {
  }
  while (!(ir.keyPressed(94))) {
  }
  rgbled_board.setColor(0, 51, 204, 0);
  rgbled_board.show();
  Serial.println("C\'est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

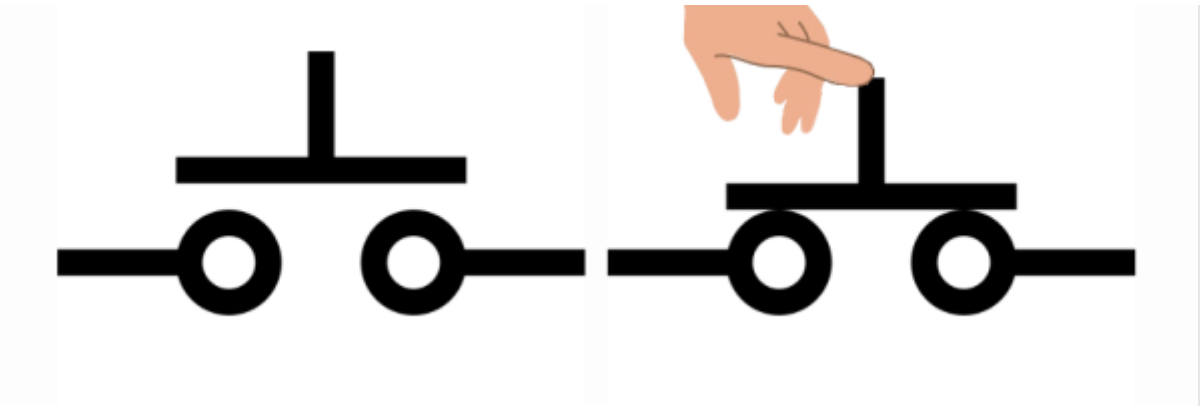
void loop() {
}
```

robot-mbot-utiliser-le-bouton

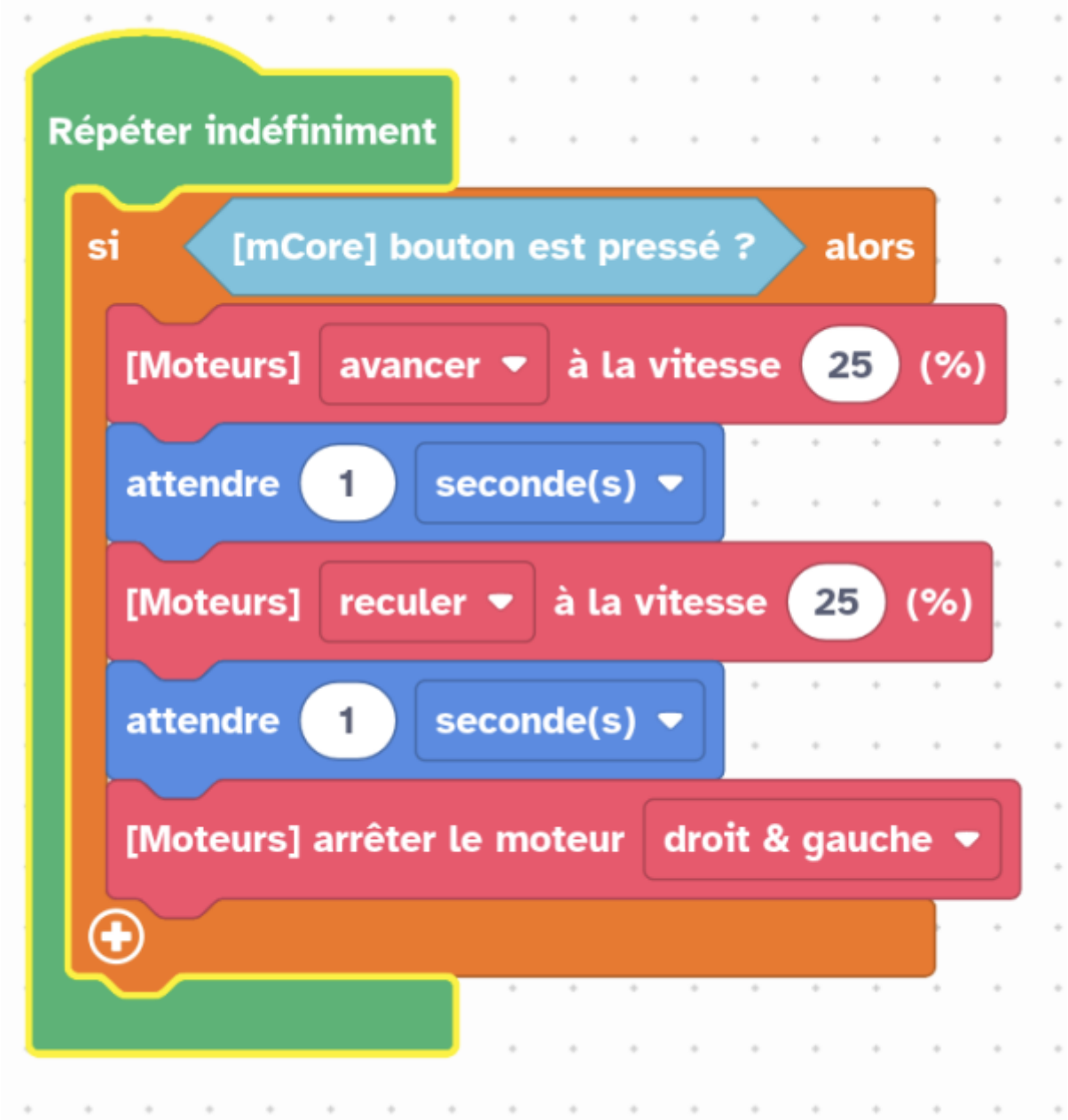
robot-mbot-utiliser-le-bouton

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));
}

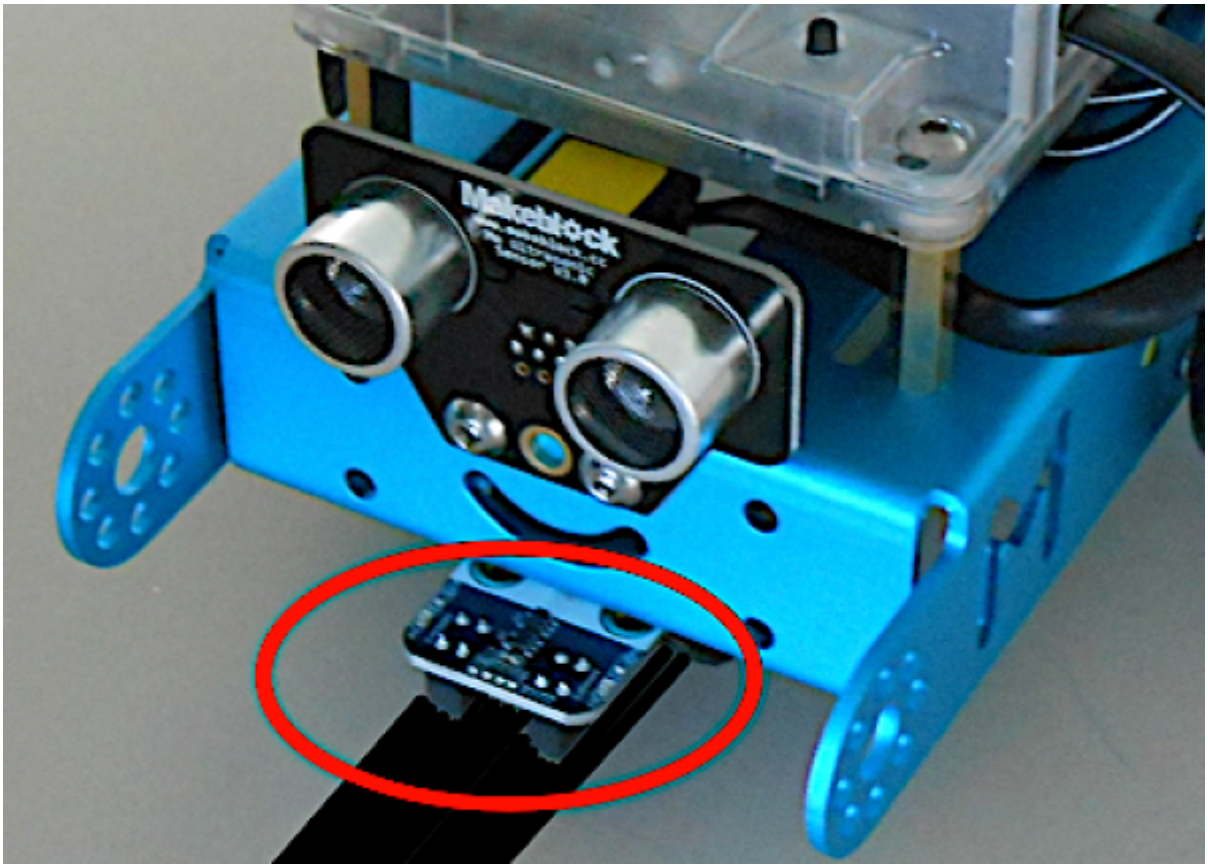

void setup() {
  pinMode(A7,INPUT);
}

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);
  }
}
```

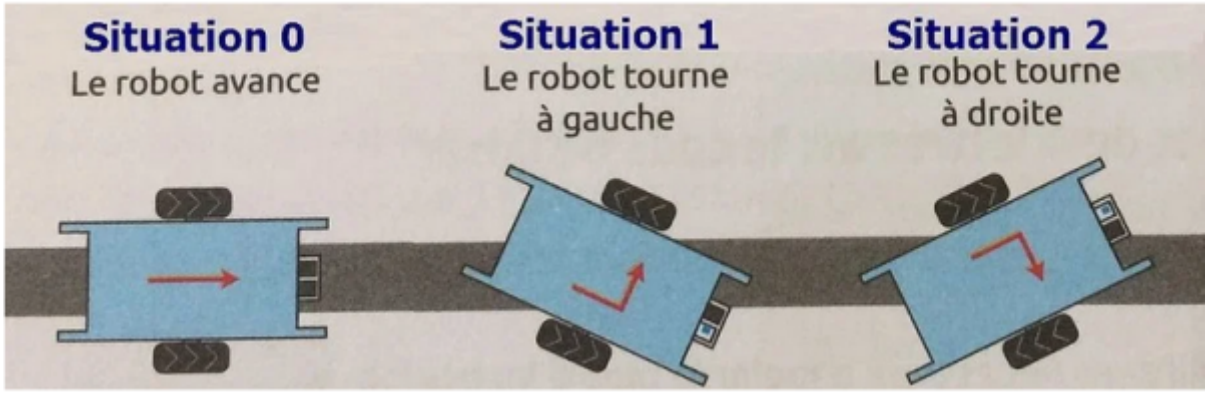
Robot suivi Ligne

robot-mbot-utiliser-les-suiveurs-de-ligne

robot_mbot_-_utiliser_les_suiveurs_de_ligne.pdf

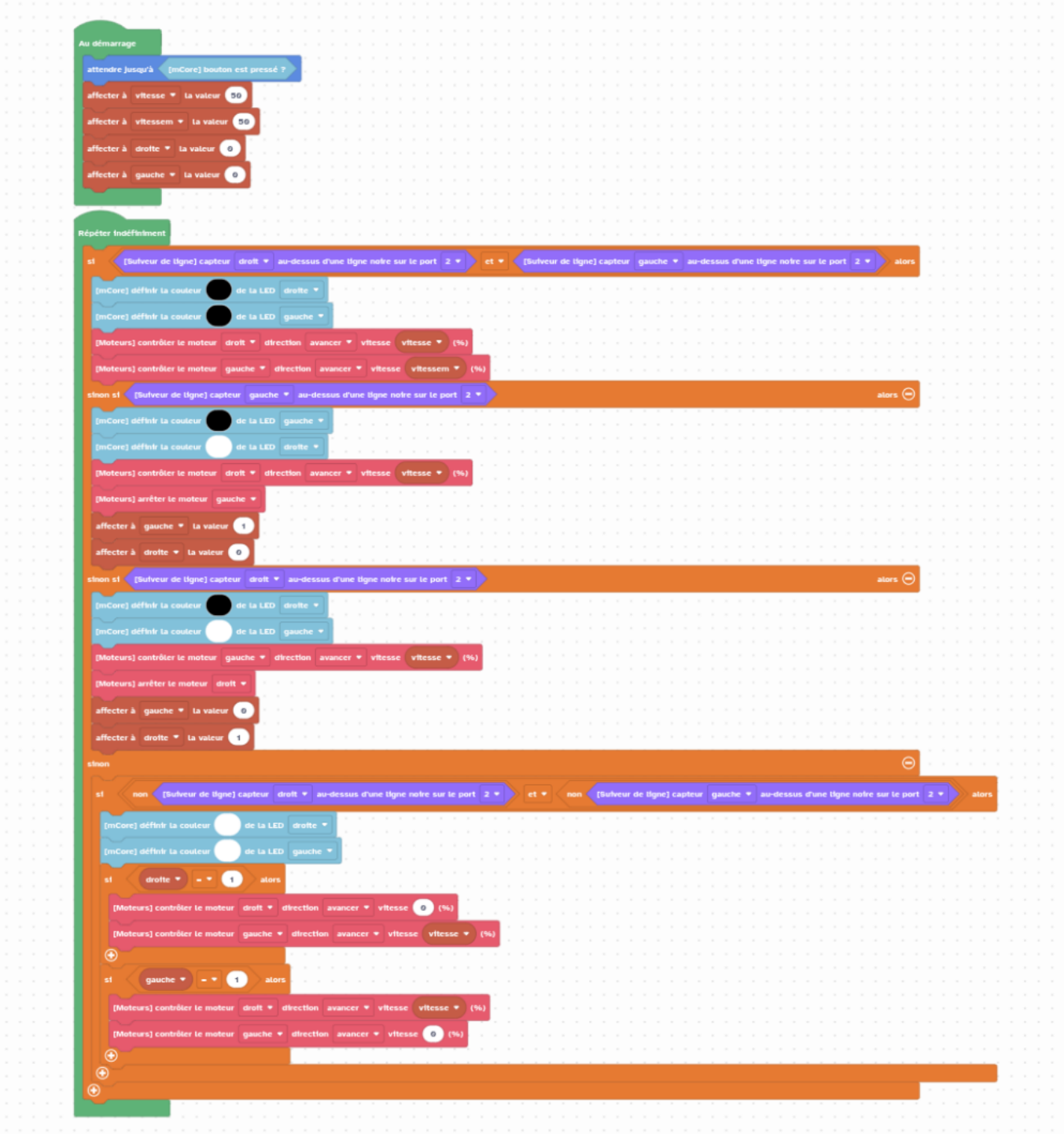






correctionsuiviligneexo4_vittascience_20251020_193057.ino.zip

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 vitessem;
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;
  vitessem = 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, vitessem);
  } 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=1794316666

Last update: **2026/11/10 14:17**