

Programme de base mBot via arduino IDE

Librairie mBot pour IDE arduino

makeblock-libraries.zip

Programme de base mBot pour test via IDE Arduino

Fonctionnalités :

4 modes disponibles :

1. mode A : permet un contrôle manuel du robot grâce aux flèches de la télécommande IR
2. mode B : le robot se déplace tout seul en évitant les obstacles grâce à un capteur à ultrasons
3. mode C : le robot suit une ligne grâce à deux capteurs de ligne
4. mode R (réglage) : Permet de choisir la vitesse voulue grâce aux chiffres de la télécommande

basembot.ino

```
//importation de la bibliothèque (library)
#include <MeMCore.h>
//Déclaration des variables
MeDCMotor motorD(M2); //moteur droit
MeDCMotor motorG(M1); //moteur gauche
MeIR ir; //capteur infrarouge
MeBuzzer buzzer;
MeUltrasonicSensor ultraSensor(PORT_3); //capteur à ultrasons
MeRGBLed led(PORT_7); //DELs multicolores (RVB)
MeLineFollower lineFinder(PORT_2); //capteurs suiveurs de ligne
int vitesse=100, distance, coef[4][2]={{1,-1},{1,1},{-1,-1},{-1,1}},
chiffre[]={22,12,24,94,8,28,90,66,82,74}, i, touche,
note[]={2637, 1975, 2093, 2349, 2093, 1975, 1760, 1760, 2093, 2637,
2349, 2093, 1975, 1975, 2093, 2349, 2637, 2093, 1760, 1760},
dureenote[]={2,1,1,2,1,1,2,1,1,2,1,1,2,1,1,2,2,2,2,0};
char mode='A';
boolean obstacle=TRUE;

void setup()
{
  ir.begin(); //lancement de la fonction de réception infrarouge
  for (i=0; i<20; i++)
  {
    buzzer.tone(note[i], 50);
    delay(dureenote[i]*50);
  }
}
```

```
void loop()
{
    distance = ultraSensor.distanceCm(); //distance séparant le robot
d'un obstacle en cm
    if (distance<=7 and not(obstacle)) {obstacle=TRUE; dels(255);}
    else if (distance>7 and obstacle) {obstacle=FALSE; dels(0);}
    if (ir.keyPressed(69)) {mode='A'; buzzer.tone(2000, 50); dels(0);}
//si la touche 'A' est pressée alors le mode 'A' est sélectionné
    if (ir.keyPressed(70)) {mode='B'; buzzer.tone(2000, 50);} //si la
touche 'B' est pressée alors le mode 'B' est sélectionné
    if (ir.keyPressed(71)) {mode='C'; buzzer.tone(2000, 50); dels(0);}
//si la touche 'C' est pressée alors le mode 'C' est sélectionné
    if (ir.keyPressed(21)) {mode='R'; déplacement(4); buzzer.tone(2000,
50); dels(0);} //si la touche 'engrenage' est pressée alors le mode 'R'
(réglage) est sélectionné

    if (mode=='A') //mode 'A' ou manuel
    {
        if (ir.keyPressed(64) and distance>7) {déplacement(0);} //si ↑ est
pressée alors avancer
        else if (ir.keyPressed(7)) {déplacement(1);} //si ← est pressée
alors tourner à gauche
        else if (ir.keyPressed(9)) {déplacement(2);} //si → est pressée
alors tourner à droite
        else if (ir.keyPressed(25)) {déplacement(3);} //si ↓ est pressée
alors reculer
        else {déplacement(4);} //sinon s'arrêter
    }
    else if (mode=='B') //mode 'B' ou évitement d'obstacles
    {
        if ((distance > 20 and obstacle==FALSE) or distance > 40) //s'il la
distance est supérieure à 20 cm et qu'il n'y avait pas d'obstacle, ou si
la distance est supérieure à 40 cm
        {
            if (obstacle){delay(50000/vitesse); dels(0);} //laisse le robot
tourner lorsqu'il y avait un obstacle (la durée dépend de la vitesse)
            déplacement(0); //avancer
            obstacle=FALSE; //pas d'obstacle
        }
    }
    else
    {
        if (not(obstacle)){dels(255);}
        buzzer.tone(2000, 50);
        déplacement(1); //tourner à gauche
        obstacle=TRUE; //présence d'un obstacle
    }
}
else if (mode=='C') //mode 'C' ou suiveur de ligne
```

```
{
  if (distance>7) {deplacement(lineFinder.readSensors());}
  else {deplacement(4);}
}
else if (mode=='R') //mode 'R' ou réglage de vitesse
{
  touche=ir.getCode();
  if (ir.decode() and touche!=0 and touche!=21) //si un appui est
détecté
  {
    for (i=0; i<10; i++) //boucle pour i allant de 0 à 10 exclu, avec
i s'incrémentant de 1 à chaque tour
    {
      if (touche==chiffre[i]){vitesse=(i+1)*25;}
    }
    mode='A';
    buzzer.tone(3000, 150);
  }
}
}

void deplacement(int X) //déclaration de la fonction 'deplacement'
{
  if (X!=4)
  {
    motorD.run(coef[X][0]*vitesse);
    motorG.run(coef[X][1]*vitesse);
  }
  else
  {
    motorD.stop();
    motorG.stop();
  }
}

void dels(int X)
{
  led.setColorAt(0,X,0,0);
  led.setColorAt(1,X,0,0);
  led.show();
}
```

From:

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

Permanent link:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:arduino:mbot:base&rev=1754471145>

Last update: 2025/08/06 11:05

