# UNIVERSITE DON BOSCO DE LUBUMBASHI

# (U.D.B.L)

FACULTE DES SCIENCES INFORMATIQUES

---


## TRAVAIL PRATIQUE DE ROBOTIQUE

---

SUJET: « **ROBOT EXPLORA** »

Dirigé par : Mr Herbert KALONJI &
Mr Baudouin BANZA

Promotion : L4 TLC

ANNEE ACADEMIQUE 2023 - 2024

# LISTE DES MEMBRES

BALIBAWA INAMWESA Agnes
BANZA KABAMBA Josué
IMONDA NDJOLI Easter
KALIBONGO HANGI Raoul
KATUNGA KAZADI Jenovic
KAZADI KALAMBAY Josué
KINAMA MAMBWE Magalie
KITO KISIMBA Mike
LIKOBELE KITENGE Jeremie
LUFATAKI MULENDA Godwin (C)
LUHINDA MUGALU Jonathan
LUKOJI BALOJI Jean-Marc
LWABA NDJIBU Kevin
MBAMBI MAVUNGU Daniel
MBAYO WA NKULU Guy
MIJI KAPOMA Majoie

# GROUPEMENT DES PARTIES

1) CHÂSSIS
- BALIBWA INAMWESA Agnes (C)
- BANZA KABAMBA Josué
- MBAMBI MAVUNGU Daniel
- KAZADI KALAMBAY Josué
- KITO KISIMBA Mike
- LWABA DJIBU Kevin

2) ENERGIE
- KATUNGA KAZADI Jenovic (C)
- KINAMA MAMBWE Magalie
- LIKOBELE KITENGE Jéremie

3) ELECTRONIQUE
- LUHINDA MUGALU Jonathan (C)
- LUKOJI BALOJI Jean-Marc
- KALIBONGO HANGI Raoul

4) PROGRAMMATION
- LUFATAKI MULENDA Godwin (C)
- MIJI KAPOMA Majoie

5) LOGISTIQUE
- MBAYO WA NKULU Guy
- IMONDA NDJOLI Easter (C)
- BALIBWA INAMWESA Agnes

***Abstract :*** *C'est dans le cadre du cours* ***d'introduction à la robotique****, que ce travail nous a été demandé, à savoir la conception d'un robot possédant un minimum de* ***quatre fonctionnalités*** *dont l'application doit se situer dans l'un des domaines suivants : domestique, médical, militaire, culinaire, de circulation routière, de service ou de mines.*

## I. INTRODUCTION

L'opinion publique a longtemps été hostile à l'égard de l'exploitation minière. Cela étant parfaitement compréhensible, vue les dommages écologiques associés à l'industrie et la destruction manifeste des mines (pollution de l'eau, perte de la biodiversité, érosion, pollution des sols, formation de puits).

Elle est non seulement néfaste pour l'environnement, mais également dangereuse pour l'exploitation humaine, au regard des mines qui sont des environnements très complexes, sombres et particulièrement imprévisibles.

Ainsi, depuis quelques années, la robotique est fréquemment mise à contribution dans ce secteur, en vue minimiser son impact négatif sur l'environnement et sur l'homme.

## II. DESCRIPTION DU PROJET

Le robot "EXPLORA" sera doté de **trois modes de fonctionnement :**

- **Mode télécommandé :** un opérateur à pourra commander le robot à distance depuis son smartphone ou son ordinateur en passant par une connexion wifi via une interface WEB
- **Mode suiveur de chemin :** dans ce mode de fonctionnement, le robot suivra un chemein bien tracé, lui permettant ainsi d'atteindre une zone bien déterminée qui a déjà été explorée ;
- **Mode autonome :** ici le robot est complètement livré à lui-même et devra esquiver les obstacles tout en s'aventurant dans la zone d'exploration.

Les fonctionnalités qui accompagneront ces modes de fonctionnement sont les suivants :

- **Capture du flux d'image en temps réel**
- **Collecte de minerais**
- **Téléguidage**
- **Détection du taux d'humidité et de la température**
- **Détection d'obstacle**
- **Suiveur de chemin**
- **Retransmission des données en temps réel**
- **Contrôle du bras robotique pour la collecte**

## MATERIELS NECESSAIRES

Le matériel sera décrit suivants les fonctionnalités

- **Quatre roues motrices :** pour la mobilité, nous utiliserons 4 roues indépendantes dotées chacune d'un moteur à engrenage (boite de vitesse) ayant un couple suffisant et ainsi garantissant une motricité assez robuste pour l'environnement minier.
- **Un pilote de moteur L298N :** offrant de bonnes caractéristiques pour piloter nos quatre moteurs ;
- **Un ESP32 :** qui constituera le cœur du robot, le contrôleur qui gérera l'entièreté du système
- **Un ESP32-CAM :** doté d'une caméra, il se chargera de la collecte d'image ainsi que de la retransmission vers l'opérateur en passant par le contrôleur.
- **Un capteur Ultrason :** qui détectera les obstacles se dressant devant
- **Quatre Capteurs de présence infrarouge :** dont
  - **Trois pour le mode suiveur de chemin**
  - **Un pour la détection du niveau du sol afin de ne pas tomber dans un trou**
  - **Et le dernier pour le niveau du bac pouvant contenir les minerais**
- **Cinq servomoteurs :** dont
  - **Quatre pour le bras**
  - **Un pour le radar constitué du capteur ultrason et de l'ESP32-CAM**
- **Des Piles 3,7V 2000mAh :** faisant office de source d'énergie et garantissant l'autonomie du robot.
- **Une plaquette perforée :** sur laquelle sera placée les différents composants
- **Une LED de 2W :** pour le l'éclairage
- **Des Jumpers :** pour la connectique
- **Des supports de microcontrôleurs mâles et femelles :** qui recevront les MCU afin de les rendre amovibles de la carte de contrôle (plaquette perforée)
- **De la colle :** pour joindre certaines parties de la maquette
- **Des vis et écrous :** pour la fixation de certains éléments du robot
- **Spray noir :** pour donner une allure suffisamment esthétique du robot Concernant la commande à distance :

- La connexion utilisée sera le **WIFI**, le robot activera un point d'accès sécurisé pour lequel seul l'opérateur pourra y accéder en usant de deux périphériques dont un qui servira de récepteur du flux vidéo provenant du robot tandis que l'autre enverra des instructions de commande pour piloter, configurer le robot.

Concernant l'application:

- Une fois connecté au point d'accès du robot, il suffit d'enter l'adresse IP par défaut du robot dans la barre de recherche d'un navigateur (ex : Chrome), pour que ce dernier nous renvoie une page d'authentification avant d'accéder à l'interface de commande et/ou de visualisation du flux vidéo

- ❖ L’interface de commande contiendra:
  - o Trois boutons pour déterminer le mode du robot
    - ▪ Mode téléguidage active 8 touches de direction (ou un analogue) pour téléguider le robot
    - ▪ Mode suiveur de chemin
    - ▪ Mode autonome
  - o Deux Boutons pour orienter l’objectif de la camera (gauche et droite) o Quatre paires de boutons pour le bras donc pour chaque servomoteur
    - ▪ Une paire pour la rotation
    - ▪ Pour la première articulation concernant l’élévation
    - ▪ Pour la seconde articulation concernant l’élévation
    - ▪ Pour la pince
  - o Un bouton pour l’ouverture et la fermeture du bac
  - o Un bouton pour stocker automatiquement l’objet saisi dans le bac o La température et le taux d’humidité du milieu dans lequel se trouve le robot

## III. OBJECTIFS

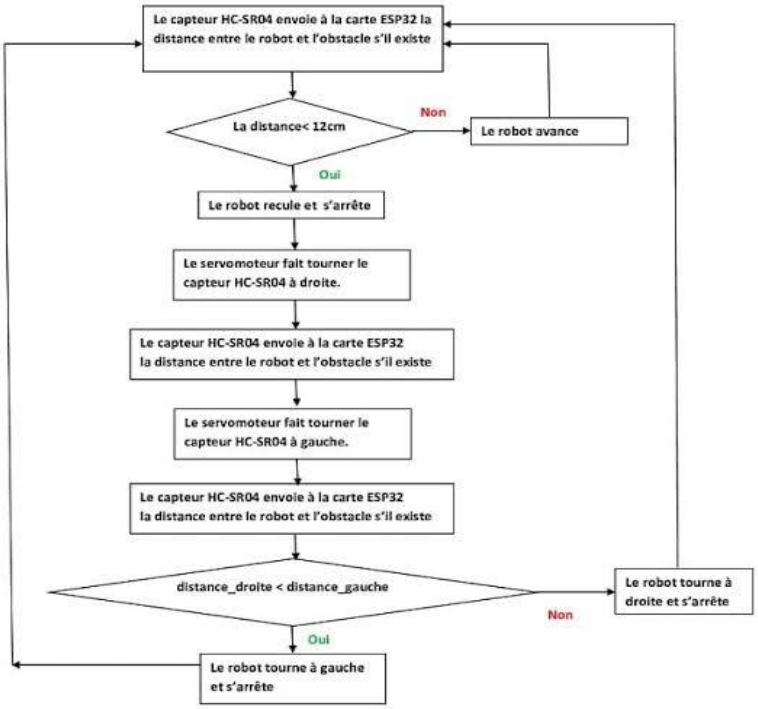
Notre projet a à cœur de proposer une solution minimisant l’impact néfaste des créations des mines en évitant les zones à faibles concentrations des minerais pouvant constituer une perte de gain, de temps et même des ressources, d’optimiser la localisation des zones riches en minerai par **l’utilisation d’un robot d’exploration des zones potentiellement minières étroites (difficile d’accès) avant d’amorcer l’exploitation (d’étendre la zone d’exploitation).**

## IV. FONCTIONNEMENT

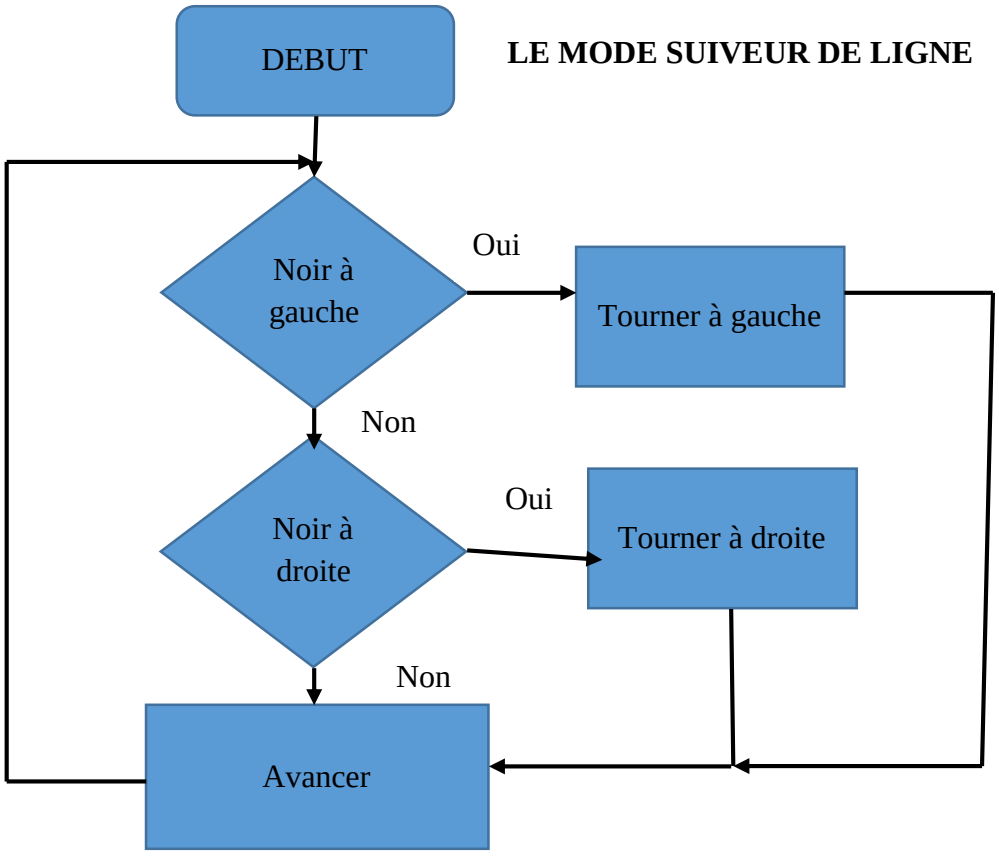
Nous allons présenter les organigrammes de deux modes de fonctionnement :

- **Le mode autonome**
- **Le mode suiveur de ligne**

## LE MODE AUTONOME




## LE MODE SUIVEUR DE LIGNE

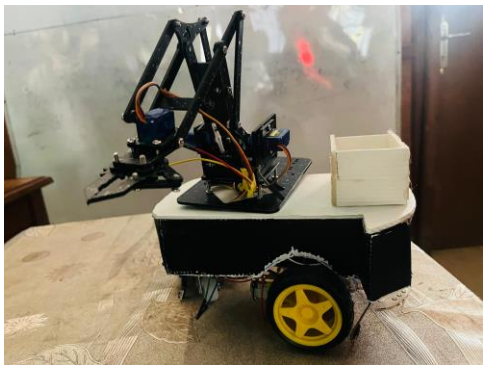


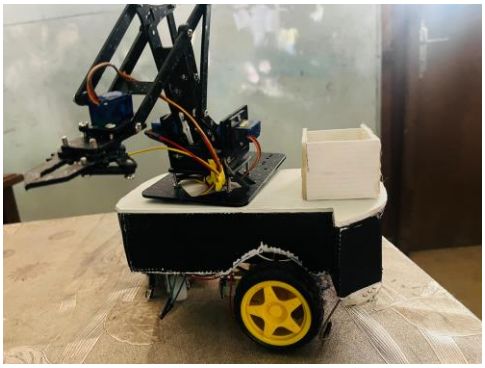
# V. CONCEPTION

## a) Présentation du châssis

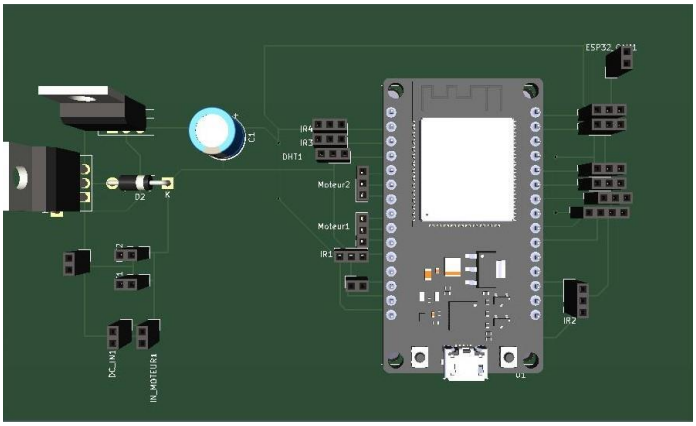
Notre châssis est principalement fait à base de :

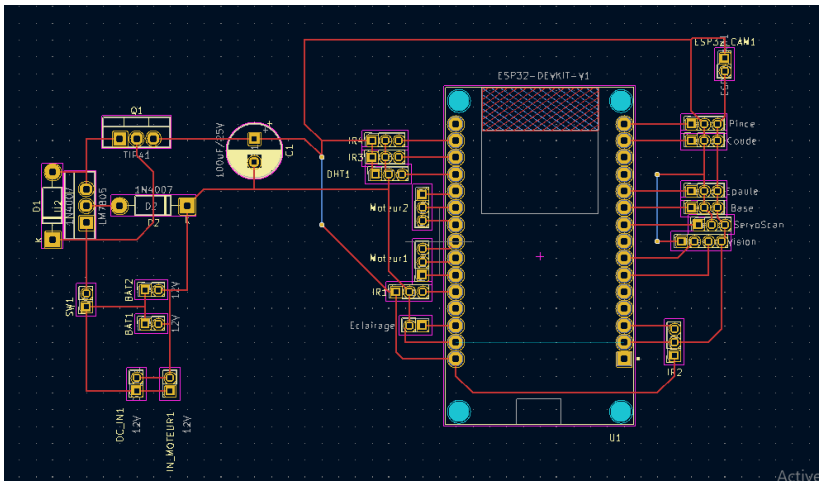
- Plastique
- Papier paille (qui est imperméable)
- Acrylique





## b) Schémas du robot



### c) Dimensionnement électrique

Pour commencer nous allons d'abord lister tous les composants pour notre robot « EXPLORA » et en suite déterminer leurs consommations électriques et en suite nous calculerons l'autonomie de notre robot en fonction des batteries à notre disposition selon la formule ci-dessous :

$$Autonomie = \frac{Energie\ totale}{Puissance\ totale}$$

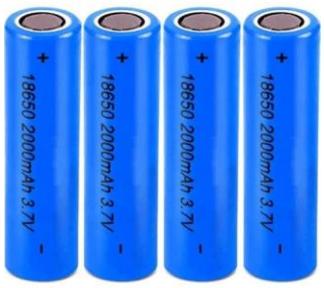
Sans oublier de retrancher les 10% pour les pertes.

| Noms composants | Nbr pieces | Voltages | Ampérages | Puissances consommées |
|---|---|---|---|---|
| **ESP32** | 1 | 5V | 170mA | 0,85W |
| **ESP32-CAM** | 1 | 5V | 2A | 10W |
| **Capt. IR** | 4 | 5V | 5mA | 0,1W |
| **L298N** | 1 | 12V | 13mA | 0,156W |
| **Led** | 1 | - | - | 2W |
| **Roues** | 4 | 6V | 200mA | 4,8W |
| **HC-SR04** | 1 | 5V | 15mA | 0,075W |
| **SG90** | 5 | 5V | 170mA | 4,25W |
| **PUISSANCE TOTALE** | | | | **22,231W** |

Batterie : 3,7V, 2000mAh

Nous allons mettre 2 blocs (de 3 batteries qui sont en série) en parallèle, ce qui nous produira une énergie de **44,4Wh**

Autonomie = 44,4Wh/22,231W = **1,99h**
Autonomie en tenant compte des pertes : **1,791h**



### d) Code

Vu le nombre des lignes de notre code, nous ne saurons pas tout mettre dans ce présent support mais nous pouvons au moins y insérer une partie.

```
#include <Arduino.h>

#include<SPIFFS.h>//Bibliothèque pour la gestion des fichiers
#include <ESPAsyncWebServer.h>//Bibliothèque pour la gestion du server WEB
#include<ArduinoJson.h>//Bibliothèque pour le formatage JSON
#include <c++/8.4.0/vector>//Bibliothèque pour la gestion des tableaux
#include <ESP32Servo.h>//Bibliothèque pour servomoteur
#include <NewPing.h>//Bibliothèque pour le capteur ultrason
#include "ESPmDNS.h"//Bibliothèque pour le service mDNS
//Bibliothèque pour le capteur DHT11
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

//=================MDNS SERVICE=======================
#define MDNS_DEVICE_NAME "explora"
#define SERVICE_NAME "EXPLORA-SERVICE"
#define SERVICE_PROTOCOL "tcp"
#define SERVICE_PORT 80

struct ServoBras
{
  Servo servo;
  int servoPin;
  String servoName;
  int initialPosition;
  int minPos;
  int maxPos;
};
```

```
//==============BROCHES UTILISEES=====================
//MOTOR A
#define IN1 26
#define IN2 25
#define EN1 27
//MOTOR B
#define IN3 32
#define IN4 33
#define EN2 15//35
//CONFIG PWM DES BROCHES
#define FREQ 5000
#define PWM_CH1 0
#define PWM_CH2 1
#define RESOLUTION 8

int speedMotor = 200;// vitesse du moteur en ligne droite
int speedMotorDev = 150;//vitesse du moteur lors des deviations/rotation

//========MODE============
#define MODE_TELEGUIDAGE 0 //TELEGUIDE
#define MODE_SUIVEUR_1 1 //SUIVEUR DE CHEMIN
#define MODE_SUIVEUR_2 2 //SUIVEUR D'OBJET//TRACKER
#define MODE_AUTONOME 3 //EVITEUR D'OBSTACLE
int exploraMode = MODE_TELEGUIDAGE;//variable du mode courant du robot
//==============SERVOMOTEUR==========
// SERVO BRAS
#define BASE 19
#define EPAULE 21
#define COUDE 22
#define PINCE  23
//Creation d'un tableau pour la gestion des servomoteurs du bras
std::vector<ServoBras> Bras =
{
  { Servo(), BASE , "Base", 90, 0, 180},
  { Servo(), EPAULE , "Epaule", 90, 0, 180},
  { Servo(), COUDE , "Coude", 90, 0, 180},
  { Servo(), PINCE , "Pince", 90, 0, 180},
};
```

```
//=========================SERVO ULTRASON ET ULTRASON================
#define ULTRASON_TRIG 17
#define ULTRASON_ECHO 16
#define DISTANCE_MAX 400 //400 CENTIMETRES
#define COU 18
NewPing vision(ULTRASON_TRIG, ULTRASON_ECHO, DISTANCE_MAX);
Servo servoScan;

//=============CAPTEURS INFRAROUGE====================================
#define BOTTOM_IR_RIGHT 34//CAPTEUR POUR LE SUIVEUR DE CHEMIN
#define BOTTOM_IR_LEFT 35//CAPTEUR POUR LE SUIVEUR DE CHEMIN
#define TOP_IR_RIGHT 39//CAPTEUR POUR LE SUIVEUR D'OBJET
#define TOP_IR_LEFT 36//CAPTEUR POUR LE SUIVEUR D'OBJET

//================CAPTEUR DE TEMPERATURE ET HUMIDITE============
#define DHT_PIN 14
#define DHTTYPE DHT11
DHT dht(DHT_PIN, DHTTYPE);
//================LED D'ECLAIRAGE================
#define LED 4
bool ledStatus = false;
//=========VARIABLES ET OBJETS POUR LA CONNECTIVITE============
const char* ssid = "EXPLORA";
const char* password = "12345678";
AsyncWebServer server(80);
AsyncWebSocket wsExplora("/wsExplora");
```

```
//========FONCTION POUR CONTROLER LES MOTEURS(LE DEPLACEMENT) DU ROBOT============

//ARRETER LE MOTEUR
void stopped(){
  ledcWrite(PWM_CH1,0);
  ledcWrite(PWM_CH2,0);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
}
void forward(){
  ledcWrite(PWM_CH1,speedMotor);
  ledcWrite(PWM_CH2,speedMotor);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}

void backward(){
  ledcWrite(PWM_CH1,speedMotor);
  ledcWrite(PWM_CH2,speedMotor);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void right(){
  ledcWrite(PWM_CH1,speedMotorDev);
  ledcWrite(PWM_CH2,speedMotorDev);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void left(){
  ledcWrite(PWM_CH1,speedMotorDev);
  ledcWrite(PWM_CH2,speedMotorDev);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
```

```
//==========FONCTION POUR LE BRAS=======================
void commandeBras(int servoIndex, int value)
{
  Bras[servoIndex].servo.write(constrain(value,Bras[servoIndex].minPos,Bras[servoIndex].maxPos));
}

//=================SUIVEUR DE LIGNE=========================
void suiveurDeLigne()
{
  int rightIR = digitalRead(BOTTOM_IR_RIGHT);
  int leftIR = digitalRead(BOTTOM_IR_LEFT);
  //Si aucun capteur ne detecte la ligne noire, avancer
  if (rightIR == LOW && leftIR == LOW)
    forward();
  //Si le capteur IR droit detecte la ligne noire, aller a droite
  else if (rightIR == HIGH && leftIR == LOW )
      right();
  //Si le capteur IR gauche detecte la ligne noire, aller a gauche
  else if (rightIR == LOW && leftIR == HIGH )
      left();
 //Si les 2 capteurs IR droit detectent la ligne noire, arreter
  else
    stopped();
}
```

```
//==========MODE AUTONOME=====================
ulong currentAuto = millis();
ulong currentAutoStop = millis();

int delayAutoStop = 200;
int delayAutoReverse = 500;
int delayAutoScan = 300;
int scanMin =0;
int scanMax = 180;
int minDetectionAuto = 30;
int etapeAuto = 0;
int distanceGauche, distanceDroite;
int distance ;

void modeAutonome()
{

  if(etapeAuto = 0)
    distance = vision.ping_cm();

  //If distance is within 30 cm then adjust motor direction as below
  if (distance > 0 && distance < minDetectionAuto || etapeAuto != 0)
  {
    //Stop motors
    if(etapeAuto == 0){//Si on detecte on arrete
      stopped();
      etapeAuto = 1;
    }
    else if(etapeAuto == 1){
      if(millis() - currentAutoStop >= delayAutoStop){//apres un temps, on recule
        backward();
        etapeAuto = 2;
      }
    }
    else if(etapeAuto == 2){//Apres un temps de recule du robot on l'arrete
      if(millis() - currentAutoStop >= delayAutoStop + delayAutoReverse){//delay(delayAutoStop)
        stopped();
        etapeAuto = 3;
      }
    }
    else if(etapeAuto == 3){
      servoScan.write(scanMin);
      etapeAuto = 4;
    }
    else if(etapeAuto == 4){
      if(millis() - currentAutoStop >= delayAutoStop + delayAutoReverse + delayAutoScan){
        distanceGauche = vision.ping_cm();
        etapeAuto = 5;
      }
    }
```

```
void modeAutonome()
    else if(etapeAuto == 5){
      servoScan.write(distanceDroite);
      etapeAuto = 6;
    }
    else if(etapeAuto == 6){
      if(millis() - currentAutoStop >= delayAutoStop + delayAutoReverse + 2 * delayAutoScan){//delay(delayAutoStop)
        distanceDroite = vision.ping_cm();
        etapeAuto = 7;
      }
    }
    else if(etapeAuto == 7){
      if (distanceGauche == 0 )
      {
        right();
      }
      else if (distanceDroite == 0 )
      {
        left();
        //delay(500);
      }
      else if (distanceGauche >= distanceDroite)
      {
        right();
        //delay(500);
      }
      else
      {
        left();
        //delay(500);
      }
      etapeAuto = 8;
      //delay(200);
    }
  }
  else if(etapeAuto == 8){
    if(millis() - currentAutoStop >= delayAutoStop + delayAutoReverse + 2 * delayAutoScan){//delay(delayAutoStop)
      stopped();
      etapeAuto = 0;
    }
  }
  else
  {
    forward();
  }
}
```

```
//==========SUIVEUR D'OBJET==================
void suiveurObjet()
{
  int MIN_DISTANCE  = 10;
  int MAX_DISTANCE  = 30;

  int distance = vision.ping_cm();
  int rightIR = digitalRead(TOP_IR_RIGHT);
  int leftIR = digitalRead(TOP_IR_LEFT);

    //Si le capteur IR gauche detecte un objet, aller a gauche
  if (rightIR == LOW && leftIR == HIGH )
  {
      left();
  }
  //Si le capteur IR droit detecte un objet, aller a droite
  else if (rightIR == HIGH && leftIR == LOW )
  {
    right();
  }
  //Si la distance entre le robot et l'objet est dans l'intervalle, avancer
  else if (distance >= MIN_DISTANCE && distance <= MAX_DISTANCE)
  {
    forward();
  }
  //sinon arreter
  else
  {
    stopped();
  }
}

//==========TELEGUIDAGE============
int commande = 0;
#define FORWARD 0
#define LEFT 1
#define RIGHT 2
#define BACKWARD 3
#define STOP 4
#define DELAY_EXPIRE 600
#define REFRESH_GUIDAGE 150

long refreshGuidage = millis(), cmdExpire = millis();
void teleguidage(){
  if(millis() - cmdExpire >= DELAY_EXPIRE)
    commande = STOP;
  if(millis() - refreshGuidage >= REFRESH_GUIDAGE){
    switch (commande)
    {
    case STOP:
```

```
//==========TELEGUIDAGE==========
int commande = 0;
#define FORWARD 0
#define LEFT 1
#define RIGHT 2
#define BACKWARD 3
#define STOP 4
#define DELAY_EXPIRE 600
#define REFRESH_GUIDAGE 150

ulong refreshGuidage = millis(), cmdExpire = millis();
void teleguidage(){
  if(millis() - cmdExpire >= DELAY_EXPIRE)
    commande = STOP;
  if(millis() - refreshGuidage >= REFRESH_GUIDAGE){
    switch (commande)
    {
    case STOP:
      stopped();
      Serial.println("=======STOP========");
      break;
    case FORWARD:
      forward();
      Serial.println("=======FORWARD========");
      break;
    case BACKWARD:
      backward();
      Serial.println("=======BACKWARD========");
      break;
    case RIGHT:
      right();
      break;
    case LEFT:
      left();
      break;

    default:
      break;
    }
  refreshGuidage = millis();
  }

}
//==============
```

```
//==============MISE A JOUR DE LA TEMPERATURE ET DE L'HUMIDITE=======================
float temperature, humidite;
void refreshDHT(){
  temperature = dht.readTemperature();
  humidite = dht.readHumidity();
}
void wifiConnexion(){
  WiFi.mode(WIFI_AP);
  if (!WiFi.softAP(ssid, password)){
    Serial.println("AP Failed");
  }
  // Config IP statique
  IPAddress AP_LOCAL_IP(192, 168, 100, 1);
  IPAddress AP_GATEWAY_IP(192, 168, 100, 1);
  IPAddress AP_NETWORK_MASK(255, 255, 255, 0);
  if (!WiFi.softAPConfig(AP_LOCAL_IP, AP_GATEWAY_IP, AP_NETWORK_MASK)) {
    Serial.println("AP Config Failed");
  }
  Serial.print("ADRESSE IP: ");
  Serial.println(WiFi.softAPIP());
  if(!MDNS.begin(MDNS_DEVICE_NAME)) {
      Serial.println("Error encountered while starting mDNS");
      return;
  }
  else{
      Serial.println("mDNS success!!!!");
  }

  MDNS.addService(SERVICE_NAME, SERVICE_PROTOCOL, SERVICE_PORT);
}
```

## VI. LIMITE

- Il ne peut pas travailler plus de deux heures sans être rechargé.
- La portée de commande du robot.
- Avec son bras il peut pas soulever des charges trop lourdes.
- Ce robot a été conçu dans le cadre du cours de robotique.

## VII. CONCLUSION

Grâce à ce travail, nous avons pu mettre en pratique les notions théoriques que nous avons apprises dans le cours de robotique et nous sommes parvenus à réaliser un prototype de robot avec les moyens à notre portée. Nous croyons que le robot « EXPLORA » peut vraiment être utile pour l'exploitation minière car c'est ce pourquoi il a été conçu et réalisé.