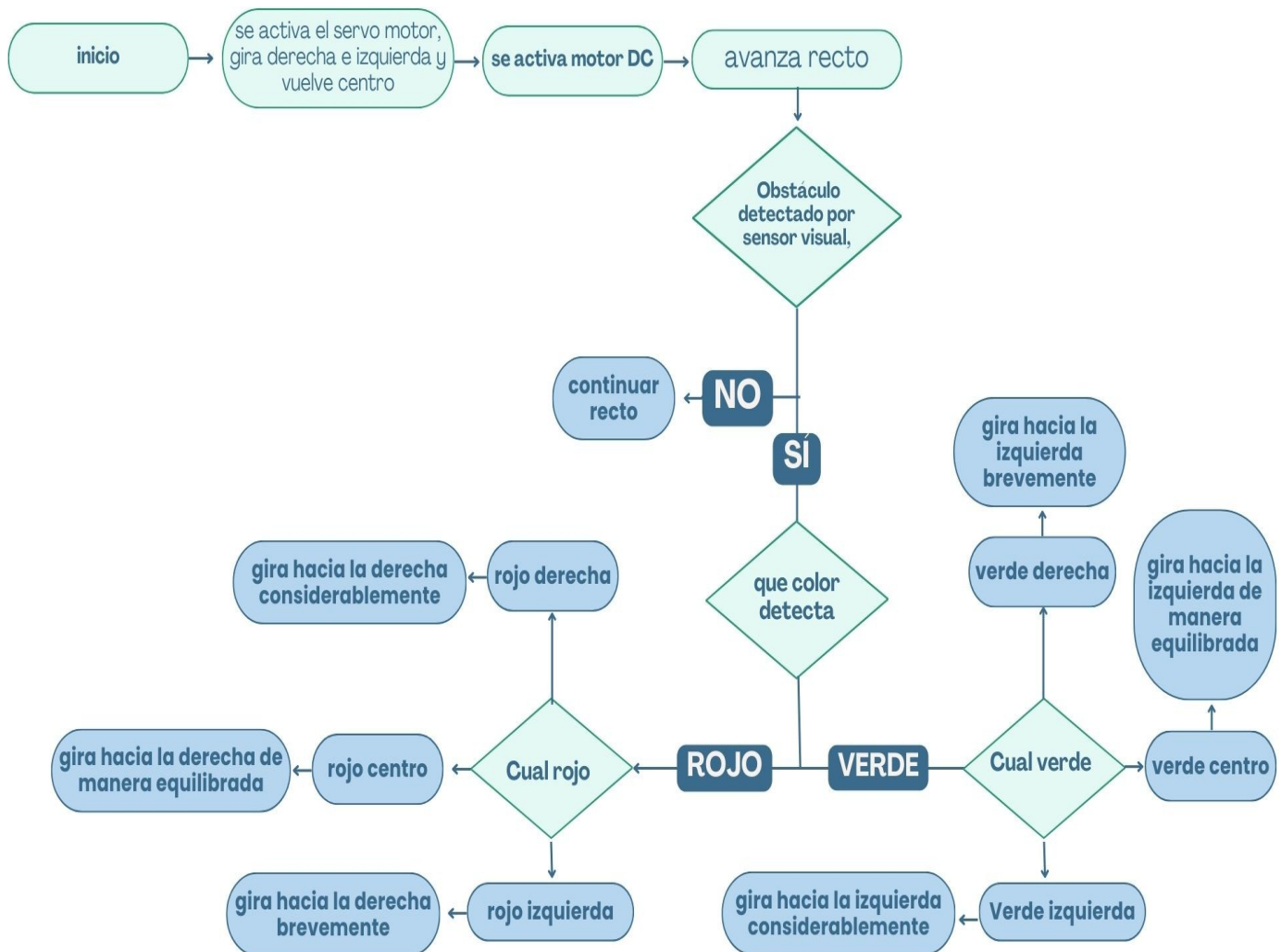


Gestión de obstáculos

Nosotros utilizamos la cámara para obtener los datos de que color es, mediante ID, cada color tiene una ID distinta, utilizamos dos tonalidades por color, siendo aprendida 4 ID, dos para cada color, siendo ID1 y rojo oscuro, ID2 verde oscuro, ID3 verde claro, ID4 rojo claro, Esto ya que la cámara a los laterales casi extremos tiene un efecto de ver mas oscuros los objetos, con cada ID tiene un proceso distinto este siendo previamente calculado ejecutado para obtener un buen desempeño.

DIAGRAMA DE FLUJO REXBOT2.0



CÓDIGO FUENTE

```
#include <NewPing.h> // Ultrasonic library
#include <Servo.h> // Servo library
Servo myservo; //servo control

#include "HUSKYLENS.h"
#include "SoftwareSerial.h"

int ID0 = 0; //not learned results. Grey result on HUSKYLENS screen
int ID1 = 1; //first learned results. colored result on HUSKYLENS screen
int ID2 = 2; //second learned results. colored result on HUSKYLENS screen

#define HUSKY_SERIAL Serial1
HUSKYLENS huskylens;
//HUSKYLENS green line >> Pin 13; blue line >> Pin 12
void printResult(HUSKYLENSResult result);

#define ENA 5
#define IN1 8 //driver Pins
#define IN2 7

int carSpeed = 80;
int carSpeedC = 115;

#define TRIGGER_PIN A0 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN A1 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define TRIGGER_PIN2 A2 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN2 A3 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define TRIGGER_PIN3 A4 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN3 A5 // Arduino pin tied to echo pin on the ultrasonic sensor.

#define MAX_DISTANCE 400// Maximum distance we want to ping for (in centimeters).
Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and
maximum distance.
NewPing lsonar(TRIGGER_PIN2, ECHO_PIN2, MAX_DISTANCE); // NewPing setup of pins and
maximum distance.
NewPing Rsonar(TRIGGER_PIN3, ECHO_PIN3, MAX_DISTANCE); // NewPing setup of pins and
maximum distance.

int rightDistance = 0, leftDistance = 0, middleDistance = 0;
```

```

int getDistance() {
delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms
should be the shortest delay between pings.
  Serial.print("Ping: ");
  Serial.print(sonar.ping_cm()); // Send ping, get distance in cm and print result (0 =
outside set distance range)
  Serial.println("cm");
}

int tilin = 0;
int grasa = 0;

void forward(){

  analogWrite(ENA, carSpeed); //enable L295n A channel
  digitalWrite(IN1,HIGH); //set IN1 hight level
  digitalWrite(IN2,LOW); //set IN2 low level
  Serial.print("Forward- "); //send message to serial monitor
  Serial.println(middleDistance);
}

void back(){
  analogWrite(ENA,170);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  Serial.print("Back");
}
void stop(){
  analogWrite(ENA,LOW);
  digitalWrite(IN1,LOW); //set IN1 low level
  digitalWrite(IN2,LOW); //set IN2 low level
}
void izquierda(){//HORARIO
  stop();
  delay(300);
  back();
  myservo.write(101);
  delay(300);
  myservo.write(71); // Cerca, ángulo 0
  analogWrite(ENA,170);
  delay(500);
  stop();
  delay(400);
  myservo.write(99);
  back();
  delay(300);
}

```

```

stop();
delay(400);
    myservo.write(80);
delay(200);
myservo.write(102);
delay(102);
}
void derecha(){//ANTIHORARIO
    back();
    myservo.write(95);
delay(280);
    myservo.write(131);    // Cerca, ángulo 0
    analogWrite(ENA,135);
delay(480);
    stop();
delay(800);
    myservo.write(95);
    back();
delay(300);
stop();
delay(400);
    myservo.write(70);
delay(200);
//myservo.write(95);
//delay(100);
}

void setup() {
    Serial.begin(115200);
    HUSKY_SERIAL.begin(9600);
    while (!huskylens.begin(HUSKY_SERIAL)) //if huskylens is not conected or the serial
of the camera is not 9600
    {
        Serial.println(F("Begin failed!"));
        Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS (General
Settings>>Protocol Type>>Serial 9600)"));
        Serial.println(F("2.Please recheck the connection."));
        delay(100);
    }
    myservo.attach(3);
    pinMode(IN1,OUTPUT);//before using io pin, pin mode must be set first
    pinMode(IN2,OUTPUT);
    pinMode(ENA,OUTPUT);
}

```

```

int pepe = 0; //its the times that the car dodge a wall

void loop(){

    analogWrite(ENA, carSpeed);
    if (pepe==0){ //beginning
        myservo.write(70);
        delay(1000);
        myservo.write(95);
        pepe++; //to do just one time
    }
    Serial.println(pepe);

    middleDistance = sonar.ping_cm(); //ultrasonic is on
    leftDistance = lsonar.ping_cm();
    rightDistance = Rsonar.ping_cm();

    forward();
    Serial.println(middleDistance);
    if (middleDistance <= 15 & middleDistance > 1){

        wall();
    }
    //pepe++;
    if (huskylens.request()) //request all blocks and arrows from
HUSKYLENS
    {
        Serial.println("#####");
        Serial.println(String()+F("Count of learned IDs:")
+huskylens.countLearnedIDs()); //The count of (faces, colors, objects or lines) you
have learned on HUSKYLENS.
        Serial.println(String()+F("frame number:")+huskylens.frameNumber()); //The number
of frame HUSKYLENS have processed.

        Serial.println("#####");
        Serial.println(String()+F("Get all blocks and arrows. Count:")+huskylens.count());
        for (int i = 0; i < huskylens.count(); i++)
        {
            HUSKYLENSResult result = huskylens.get(i);
            printResult(result);
        } //the camera identificate the object and execute a routine

    }
}

```

```

else{
    Serial.println("Fail to request objects from Huskylens!");
}

delay(50);

}

void printResult(HUSKYLENSResult result){
    if (result.command == COMMAND_RETURN_BLOCK){//result is a block
        Serial.println(String()+F("Block:xCenter=")+result.xCenter+F(",yCenter=")
+result.yCenter+F(",width=")+result.width+F(",height=")+result.height+F(",ID=")
+result.ID);
        Serial.println(result.ID);
    }
    else if (result.command == COMMAND_RETURN_ARROW){//result is an arrow
        Serial.println(String()+F("Arrow:xOrigin=")+result.xOrigin+F(",yOrigin=")
+result.yOrigin+F(",xTarget=")+result.xTarget+F(",yTarget=")+result.yTarget+F(",ID=")
+result.ID);
    }
    //ID1, ID4 DETECCION ROJO
    if (result.ID == 1 && result.ID == 4 && result.height>70 && result.xOrigin >= 188) {
        Serial.println("-----red -----");//avoid for right
        stop();
        delay(400);
        back();
    delay(300);
    stop();
    delay(200);
        forward();
        analogWrite(ENA,carSpeedC);
        myservo.write(125);
        delay(600);//PRIMER GIRO DESPUES DE DETECTAR ROJO
        Serial.println("right");

        myservo.write(73);
        delay(1200);//CORRECCION LUEGO DEL PRIMER GIRO(MOVIENTO IZQUIERDA)
        Serial.println("left");
        myservo.write(115);
        delay(400);//DELAY ENDERESAMIENTO

        myservo.write(101);//centro
    }
    //FLANCO DERECHO
    else if (result.ID == 1 && result.height>70 && result.xOrigin >= 188) {
        Serial.println("-----red -----");//avoid for right
        stop();
    }

```

```

    delay(400);
    back();
delay(300);
stop();
delay(400);
    forward();
    analogWrite(ENA, carSpeedC);
    myservo.write(125);
    delay(550); //PRIMER GIRO DESPUES DE DETECTAR ROJO
    Serial.println("right");

    myservo.write(73);
    delay(1200); //CORRECCION LUEGO DEL PRIMER GIRO(MOVIENTO IZQUIERDA)
    Serial.println("left");
    myservo.write(110);
    delay(600); //DELAY ENDERESAMIENTO

    myservo.write(95); //centro
    stop();
    delay(400);
}

//FLANCO IZQUIERDA
else if (result.ID == 1 && result.ID == 4 && result.height > 70 && result.xOrigin <
188){
    Serial.println("----- low led red -----"); //avoid less for right
    stop();
    //delay(400);
    //back();
    //delay(300);
    //stop();
    delay(200);
    forward();
    analogWrite(ENA, carSpeedC );
    myservo.write(131);
    delay(600);
    Serial.println("right");

    myservo.write(73);
    delay(700); //delay enderezamiento
    Serial.println("left");
    myservo.write(115);
    delay(400);

    myservo.write(95); //centro
    stop();

```

```

    delay(400);
}

else if (result.ID == 1 && result.height>70 && result.xOrigin < 188){
    Serial.println("----- low red -----");//avoid less for right
    stop();
    delay(400);
    back();
    delay(300);
    stop();
    delay(200);
    forward();
    analogWrite(ENA,carSpeedC);
    myservo.write(125);
    delay(400);
    Serial.println("right");

    myservo.write(73);
    delay(700);
    Serial.println("left");
    myservo.write(115);
    delay(300);

    myservo.write(95);//centro
    stop();
    delay(400);
}

else if (result.ID == 4 && result.height>70 && result.xOrigin < 188){
    Serial.println("----- LED2 red -----");//avoid less for right
    stop();
    delay(400);
    back();
    delay(300);
    stop();
    delay(200);
    forward();
    analogWrite(ENA,carSpeedC);
    myservo.write(125);
    delay(500);
    Serial.println("right");

    myservo.write(80);
    delay(650);
    Serial.println("left");
    myservo.write(110);
    delay(450);
}

```



```

        myservo.write(101); //centro
        stop();
        delay(400);
    }
//VERDE IZQUIERDA
else if (result.ID == 2 && result.ID == 3 && result.height>70 && result.xOrigin <=
135){
    Serial.println("-----green with LED-----"); //avoid for left
    stop();
    delay(400);
    back();
    delay(300);
    stop();
    delay(200);
    forward();
    analogWrite(ENA, carSpeedC);
    myservo.write(80);
    delay(700);
    Serial.println("left");

    myservo.write(125);
    delay(900);
    Serial.println("right");
    myservo.write(75);
    delay(800);
    myservo.write(95); //centro
    stop();
    delay(400);
}

//VERDE IZQUIERDA

else if (result.ID == 2 && result.height>70 && result.xOrigin <= 135){ //izquierdo
    Serial.println("-----green -----"); //avoid for left
    stop();
    delay(400);
    back();
    delay(250);
    stop();
    delay(200);
    forward();
    analogWrite(ENA, carSpeed);
    myservo.write(80);
    delay(900);
    Serial.println("left");

    myservo.write(125);

```

```

    delay(950);
    Serial.println("right");
    myservo.write(75);
    delay(800);
    myservo.write(95); //centro
    stop();
    delay(400);
}
//VERDE DERECHA

else if (result.ID == 2 && result.ID == 3 && result.height>70 && result.xOrigin >
130){ //flanco derecho
    Serial.println("----- low green with LED -----"); //avoid less for
left
    stop();
    delay(400);
    back();
    delay(300);
    forward();
    analogWrite(ENA, carSpeedC);
    myservo.write(75);
    delay(400);
    Serial.println("left");

    myservo.write(125);
    delay(680);
    Serial.println("right");
    myservo.write(75);
    delay(650);

    myservo.write(95); //centro
    stop();
    delay(400);

}
//VERDE DERECHAA
else if (result.ID == 2 && result.height>70 && result.xOrigin > 130){
    Serial.println("----- low green -----"); //avoid less for left
    stop();
    delay(400);

    forward();
    analogWrite(ENA, carSpeedC);
    myservo.write(75);
    delay(650);
    Serial.println("left");

```

```

    myservo.write(125);
    delay(700);
    Serial.println("right");
    myservo.write(75);
    delay(700);

    myservo.write(95);//centro
    stop();
    delay(400);

}

//VERDE medio
else if (result.ID ==3  && result.height>70 && result.xOrigin > 130){
    Serial.println("----- led green -----"); //avoid less for left
    stop();
    delay(400);
    back();
    delay(300);

    forward();
    analogWrite(ENA,0);
    myservo.write(75);
    delay(500);
    Serial.println("left");

    myservo.write(125);
    delay(550);
    Serial.println("right");
    myservo.write(75);
    delay(350);

    myservo.write(95);//centro
    stop();
    delay(400);

}

else if (result.ID == 5 && result.height>70 && pepe > 2){
    //parking in magenta when the car  dodge 15 walls
    Serial.println("MAGENTA");
    back();
    delay(700);

    myservo.write (100);
    forward();

```

```

    delay(1000);

    myservo.write (99);
    back();

    delay(1000);
    stop();
    forward();
    delay(500);
    stop();
    delay(400);

}

}

void wall(){

    stop();
    delay(300);
    analogWrite(ENA, carSpeed);
    Serial.println(middleDistance);
    pepe++;
    if (leftDistance <= 70 && leftDistance > 1 && grasa == 0){
        tilin++;
    }
    else if (tilin == 0){
        grasa++;
    }

    Serial.println(leftDistance); // if leftdistance > 50 = clockwise
    //if (leftDistance <= 80 && leftDistance > 1){
    //if (digitalRead(41)){
    if (tilin > 0 && grasa == 0){

        izquierda();

    }

    //sin pin antihorario
    else if (grasa > 0){
        derecha();
        myservo.write(94);
    }
}

```

```
if (pepe>13){  
  delay(1);  
  stop();  
  delay(1000000000);  
}  
} //fin-wall
```

Aquí les proporcionamos un link para que pueda actualizar el firmware y como aprender a utilizar la IA de la camara = <https://learn.dfrobot.com/makelog-308564.html>