//MOTORS

#include <Servo.h>

Servo myservo;

int inputPin = A0; // ultrasonic module ECHO to A0

int outputPin = A1; // ultrasonic module TRIG to A1

#define Lpwm\_pin 5 //pin of controlling speed---- ENA of motor driver board

#define Rpwm\_pin 6 //pin of controlling speed---- ENB of motor driver board

int pinLB = 2; //pin of controlling turning---- IN1 of motor driver board

int pinLF = 4; //pin of controlling turning---- IN2 of motor driver board

int pinRB = 7; //pin of controlling turning---- IN3 of motor driver board

int pinRF = 8; //pin of controlling turning---- IN4 of motor driver board

unsigned char Lpwm\_val = 100; //initialized left wheel speed at 250

unsigned char Rpwm\_val = 100; //initialized right wheel speed at 250

int Car\_state = 0; //the working state of car

int myangle; //defining variable of angle

int pulsewidth; //defining variable of pulse width

unsigned char DuoJiao = 90; //initialized angle of motor at 90°

//COLOR SENSOR

#define S0 9

#define S1 10

#define S2 12

#define S3 13

#define sensorOut 11

//Output from the sensor:

int redFrequency = 0;

int greenFrequency = 0;

int blueFrequency = 0;

//Formatted color values:

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

//Values used for calibration

int redMin;

int redMax;

int greenMin;Sv

int greenMax;SS

int blueMin;

int blueMax;

int color = 0;

void M\_Control\_IO\_config(void)

{

pinMode(pinLB, OUTPUT); // /pin 2

pinMode(pinLF, OUTPUT); // pin 4

pinMode(pinRB, OUTPUT); // pin 7

pinMode(pinRF, OUTPUT); // pin 8

pinMode(Lpwm\_pin, OUTPUT); // pin 5 (PWM)

pinMode(Rpwm\_pin, OUTPUT); // pin 6(PWM)

}

void Set\_Speed(unsigned char Left, unsigned char Right) //function of setting speed

{

analogWrite(Lpwm\_pin, Left);

analogWrite(Rpwm\_pin, Right);

}

void advance() // going forward

{

digitalWrite(pinRB, LOW); // making motor move towards right rear

digitalWrite(pinRF, HIGH);

digitalWrite(pinLB, LOW); // making motor move towards left rear

digitalWrite(pinLF, HIGH);

Car\_state = 1;

}

void back() //back up

{

digitalWrite(pinRB, HIGH); //making motor move towards right rear

digitalWrite(pinRF, LOW);

digitalWrite(pinLB, HIGH); //making motor move towards left rear

digitalWrite(pinLF, LOW);

Car\_state = 2;

}

void turnR() //turning right(dual wheel)

{

digitalWrite(pinRB, LOW); //making motor move towards right rear

digitalWrite(pinRF, HIGH);

digitalWrite(pinLB, HIGH);

digitalWrite(pinLF, LOW); //making motor move towards left front

Car\_state = 4;

}

void turnL() //turning left(dual wheel)

{

digitalWrite(pinRB, HIGH);

digitalWrite(pinRF, LOW ); //making motor move towards right front

digitalWrite(pinLB, LOW); //making motor move towards left rear

digitalWrite(pinLF, HIGH);

Car\_state = 3;

}

void stopp() //stop

{

digitalWrite(pinRB, HIGH);

digitalWrite(pinRF, HIGH);

digitalWrite(pinLB, HIGH);

digitalWrite(pinLF, HIGH);

Car\_state = 5;

}

void Self\_Control(void)//self-going, ultrasonic obstacle avoidance

{

int H;

myservo.write(DuoJiao);

H = Ultrasonic\_Ranging(1);

delay(300);

if (Ultrasonic\_Ranging(1) < 35)

{

stopp();

delay(100);

back();

delay(300);

}

if (Ultrasonic\_Ranging(1) < 50)

{

stopp();

delay(100);

myservo.write(0);

int L = ask\_pin\_L(2);

delay(300);

myservo.write(180);;

int R = ask\_pin\_R(3);

delay(300);

//color sensor loop

readColor();//read sensor

decideColor();//format color values

// printColor();//print values

//-----------------

if (color == 0) {

Serial.println("COLOR: BLANCO NEGRO O DESCONOCIDO");

if (ask\_pin\_L(2) > ask\_pin\_R(3))

{

back();

delay(400);

turnL();

delay(200);

advance();

delay(500);

stopp();

delay(50);

myservo.write(DuoJiao);

H = Ultrasonic\_Ranging(1);

delay(500);

}

if (ask\_pin\_L(2) <= ask\_pin\_R(3))

{

back();

delay(400);

turnR();

delay(200);

advance();

delay(500);

stopp();

delay(50);

myservo.write(DuoJiao);

H = Ultrasonic\_Ranging(1);

delay(300);

}

} else if (color == 1) {

Serial.println("COLOR: NARANJA AMARILLO MORADO O ROJO");

back();

delay(400);

turnL();

delay(200);

advance();

delay(500);

stopp();

delay(50);

myservo.write(DuoJiao);

H = Ultrasonic\_Ranging(1);

delay(500);

} else if (color == 2) {

Serial.println("COLOR: VERDE O AZUL");

back();

delay(400);

turnR();

delay(200);

advance();

delay(500);

stopp();

delay(50);

myservo.write(DuoJiao);

H = Ultrasonic\_Ranging(1);

delay(500);

}

}

else

{

advance();

}

}

int Ultrasonic\_Ranging(unsigned char Mode)//function of ultrasonic distance detecting ，MODE=1，displaying，no displaying under other situation

{

int old\_distance;

digitalWrite(outputPin, LOW);

delayMicroseconds(2);

digitalWrite(outputPin, HIGH);

delayMicroseconds(10);

digitalWrite(outputPin, LOW);

int distance = pulseIn(inputPin, HIGH); // reading the duration of high level

distance = distance / 58; // Transform pulse time to distance

if (Mode == 1) {

Serial.print("\n H = ");

Serial.print(distance, DEC);

return distance;

}

else return distance;

}

int ask\_pin\_L(unsigned char Mode)

{

int old\_Ldistance;

digitalWrite(outputPin, LOW);

delayMicroseconds(2);

digitalWrite(outputPin, HIGH);

delayMicroseconds(10);

digitalWrite(outputPin, LOW);

int Ldistance = pulseIn(inputPin, HIGH);

Ldistance = Ldistance / 58; // Transform pulse time to distance

if (Mode == 2) {

Serial.print("\n L = ");

Serial.print(Ldistance, DEC);

return Ldistance;

}

else return Ldistance;

}

int ask\_pin\_R(unsigned char Mode)

{

int old\_Rdistance;

digitalWrite(outputPin, LOW);

delayMicroseconds(2);

digitalWrite(outputPin, HIGH); //

delayMicroseconds(10);

digitalWrite(outputPin, LOW);

int Rdistance = pulseIn(inputPin, HIGH);

Rdistance = Rdistance / 58; // Transform pulse time to distance

if (Mode == 3) {

Serial.print("\n R = ");

Serial.print(Rdistance, DEC);

return Rdistance;

}

else return Rdistance;

}

void setup()

{

//COLOR SENSOR SETUP

//Declarations:

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

pinMode(13, OUTPUT);

pinMode(sensorOut, INPUT);

// Set frequency scaling to 20%:

digitalWrite(S0, HIGH);

digitalWrite(S1, LOW);

Serial.begin(9600);//begin serial communication

calibrate();//calibrate sensor (look at serial monitor)

//--------------------

myservo.attach(A2);

M\_Control\_IO\_config(); //motor controlling the initialization of IO

Set\_Speed(Lpwm\_val, Rpwm\_val); //setting initialized speed

myservo.write(DuoJiao); //setting initialized motor angle

pinMode(inputPin, INPUT); //starting receiving IR remote control signal

pinMode(outputPin, OUTPUT); //IO of ultrasonic module

Serial.begin(9600); //initialized serial port , using Bluetooth as serial port, setting baud

stopp(); //stop

delay(1000);

}

void loop()

{

Self\_Control();

}

//COLOR SENSOR FUNTIONS

void decideColor() {//format color values

//Limit possible values:

redColor = constrain(redColor, 0, 255);

greenColor = constrain(greenColor, 0, 255);

blueColor = constrain(blueColor, 0, 255);

//find brightest color:

int maxVal = max(redColor, blueColor);

maxVal = max(maxVal, greenColor);

//map new values

redColor = map(redColor, 0, maxVal, 0, 255);

greenColor = map(greenColor, 0, maxVal, 0, 255);

blueColor = map(blueColor, 0, maxVal, 0, 255);

redColor = constrain(redColor, 0, 255);

greenColor = constrain(greenColor, 0, 255);

blueColor = constrain(blueColor, 0, 255);

//decide which color is present (you may need to change some values here):

if (redColor > 250 && greenColor > 250 && blueColor > 250) {

color = 0;//white

}

else if (redColor < 25 && greenColor < 25 && blueColor < 25) {

color = 0;//black

}

else if (redColor > 200 && greenColor > 200 && blueColor < 100) {

color = 1;//yellow

}

else if (redColor > 200 && greenColor > 25 && blueColor < 100) {

color = 1;//orange

}

else if (redColor > 200 && greenColor < 100 && blueColor > 200) {

color = 1;//purple

}

else if (redColor > 250 && greenColor < 200 && blueColor < 200) {

color = 1;//red

}

else if (redColor < 200 && greenColor > 250 && blueColor < 200) {

color = 2;//green

}

else if (redColor < 200 /\*&& greenColor < 200\*/ && blueColor > 250) {

color = 2;//blue

}

else {

color = 0;//unknown

}

}

void calibrate() {

Serial.println("Calibrating...");

Serial.println("White");//aim sensor at something white

//set calibration values:

digitalWrite(13, HIGH);

delay(2000);

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

redMin = 50 ; // pulseIn(sensorOut, LOW);

delay(100);

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

greenMin = 57;//pulseIn(sensorOut, LOW);

delay(100);

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

blueMin = 43;//pulseIn(sensorOut, LOW);

delay(100);

Serial.println("next...");//aim sensor at something black

digitalWrite(13, LOW);

delay(2000);

Serial.println("Black");

//set calibration values:

digitalWrite(13, HIGH);

delay(2000);

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

redMax = 265;// pulseIn(sensorOut, LOW);

delay(100);

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

greenMax = 340;//pulseIn(sensorOut, LOW);

delay(100);

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

blueMax = 245;//pulseIn(sensorOut, LOW);

delay(100);

Serial.println("Done calibrating.");

digitalWrite(13, LOW);

}

void readColor() {//get data from sensor

//red:

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

redFrequency = pulseIn(sensorOut, LOW);

redColor = map(redFrequency, redMin, redMax, 255, 0);

delay(100);

//green:

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

greenFrequency = pulseIn(sensorOut, LOW);

greenColor = map(greenFrequency, greenMin, greenMax, 255, 0);

delay(100);

//blue:

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

blueFrequency = pulseIn(sensorOut, LOW);

blueColor = map(blueFrequency, blueMin, blueMax, 255, 0);

delay(100);

}