INTELIGENCIA ARTIFICIAL

PROYECTO FINAL

JAIRO ANDRES SALAZAR

CRISTIAN RESTREPO

PROFESOR:

CARLOS LONDOÑO

CORPORACION DE ESTUDIOS TECOLOGICOS DEL NORTE DEL VALLE

TECNOLOGIA EN SISTEMA DE INFORMACION

CARTAGO

2018

**Carro-robot que evita obstáculos.**

Este robot puede rotar su cabeza y puede cambiar su dirección donde hay más espacio.

**Materiales:**

* Arduino Uno
* Ultrasonido
* L298N H (Puente H)
* Servo Motor
* Kit Carro
* Cables macho-macho y macho-hembra
* Breadboard
* Pila 9V
* Cargador Portátil

**Librería NewPing:**

La biblioteca NewPing agrega muchas características nuevas y da nueva vida a los sensores de distancia.

**Características:**

* Funciona con muchos modelos diferentes de sensores ultrasónicos: SR04, SRF05, SRF06, DYP- ME007 y Parallax PING))).
* Opción para interactuar con todos menos el sensor SRF06 que usa solo un pin para Arduino.
* No se demora un segundo completo si no se recibe eco de ping como todas las demás bibliotecas de ultrasonidos.
* Compatible con toda la gama Arduino (y clones), la familia Teensy (incluidos los $ 19.80 96Mhz 32 bit Teensy 3.2) y los microcontroladores no AVR.
* Hace ping a los sensores de manera consistente y confiable hasta 30 veces por segundo.
* Método de interrupción del temporizador para bocetos controlados por eventos.
* Método de filtro digital incorporado ping\_median () para una fácil corrección de errores.
* Utiliza registros de puertos al acceder a los pines para una ejecución más rápida y un tamaño de código más pequeño.
* Facilidad de uso de sensores múltiples (boceto de ejemplo con 15 sensores).
* Cálculo de distancia más preciso (cm, pulgadas y uS).
* No usa pulseIn, que es lento y da resultados incorrectos con algunos modelos de sensores ultrasónicos.
* Desarrollado activamente con funciones añadidas y errores.

**Constructor:**

NewPing sonar (trigger\_pin, echo\_pin [, max\_cm\_distance]);

**Ejemplo de configuración librería NewPing en Ultrasonido Proyecto Final Carro Lógica Difusa**

NewPing sonar (A1, A2, 200);

Esto inicializa NewPing para usar el pin A1 para la salida del disparador, el pin A2 para la entrada del eco, con una distancia máxima de ping de 200 cm. max\_cm\_distance es opcional [predeterminado = 500 cm]. Si se conecta utilizando un solo pin, especifique el mismo pin para trigger\_pin y echo\_pin, ya que el mismo pin realiza ambas funciones.

**Métodos:**

* sonar.ping (); - Enviar un ping, devuelve el tiempo de eco en microsegundos o 0 (cero) si no hay eco de ping dentro del límite de distancia establecido.
* sonar.ping\_in (); - Enviar un ping, devuelve la distancia en pulgadas o 0 (cero) si no hay eco de ping dentro del límite de distancia establecido.
* sonar.ping\_cm (); - Enviar un ping, devuelve la distancia en centímetros o 0 (cero) si no hay eco de ping dentro del límite de distancia establecido.
* sonar.ping\_median (iteraciones); - Hacer varios pings (por defecto = 5), descartar pings fuera de rango y devolver mediana en microsegundos.
* sonar.convert\_in (echoTime); - Convierte microsegundos en distancia en pulgadas.
* sonar.convert\_cm (echoTime); - Convierte microsegundos en distancia en centímetros.
* sonar.ping\_timer (función); - Enviar una función ping y llamada para probar si el ping está completo.
* sonar.check\_timer (); - Verifique si el ping ha regresado dentro del límite de distancia establecido.
* timer\_us (frecuencia, función); - Función de llamada cada microsegundos de frecuencia.
* timer\_ms (frecuencia, función); - Función de llamada cada milisegundos de frecuencia.
* timer\_stop (); - Detener el temporizador.

**Código:**

#include <Servo.h> //Libreria servo motor. Esta es una biblioteca estándar

#include <NewPing.h> //Libreria de funciones del sensor ultrasónico. Debe instalar esta libreria

//pines de control L298N (puente H)

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 4;

const int RightMotorBackward = 5;

//pines de sensor

#define trig\_pin A1 //entrada analoga 1

#define echo\_pin A2 //entrada analoga 2

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

//funcion del sensor

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance);

//nombre del servo

Servo servo\_motor;

void setup(){

pinMode(RightMotorForward, OUTPUT);

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

//pin del servo

servo\_motor.attach(10);

servo\_motor.write(115);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop(){

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20){

moveStop();

delay(300);

moveBackward();

delay(400);

moveStop();

delay(300);

distanceRight = lookRight();

delay(300);

distanceLeft = lookLeft();

delay(300);

if (distance >= distanceLeft){

turnRight();

moveStop();

}

else{

turnLeft();

moveStop();

}

}

else{

moveForward();

}

distance = readPing();

}

int lookRight(){

servo\_motor.write(50);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

}

int lookLeft(){

servo\_motor.write(170);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(115);

return distance;

delay(100);

}

int readPing(){

delay(70);

int cm = sonar.ping\_cm();

if (cm==0){

cm=250;

}

return cm;

}

void moveStop(){

digitalWrite(RightMotorForward, LOW);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

digitalWrite(LeftMotorBackward, LOW);

}

void moveForward(){

if(!goesForward){

goesForward=true;

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

}

void moveBackward(){

goesForward=false;

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorBackward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

**Video Funcionamiento del Carro evadiendo obstáculos.**

<https://www.youtube.com/watch?v=fF35_APdexk&feature=em-share_video_user>