// CODIGO DESVIA OBSTACULO PARA O EasyBotDS2

// Made by Mauricio Duarte

// PINAGEM

// Servo da Cabeça no pino 11

// Servo esteira direita no pino 12

// Servo esteira esquerda no pino 13

#include <NewPing.h>

#include <Servo.h>

Servo servo\_esteira\_dir;

Servo servo\_esteira\_esq;

Servo servo\_cabeca;

void ParadaTotal();

void VirarDireita();

void VirarEsquerda();

void AndarFrente();

void AndarTras();

int dir\_zero;

int dir\_max\_antihorario;

int dir\_max\_horario;

int esq\_zero;

int esq\_max\_antihorario;

int esq\_max\_horario;

#define TRIGGER\_PIN 4 // Arduino pin tied to trigger pin on ping sensor.

#define ECHO\_PIN 5 // Arduino pin tied to echo pin on ping sensor.

#define MAX\_DISTANCE 200 // 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.

unsigned int pingSpeed = 40; // How frequently are we going to send out a ping (in milliseconds). 50ms would be 20 times a second.

unsigned long pingTimer; // Holds the next ping time.

int Distance = 200; // Recebe a leituta de distacia do sensor

int triggerDistance = 30; // Distancia para o robô parar e procurar outro caminho

int TempoGiro = 1700; // Tempo para controlar o giro de 90 graus

void setup() {

 Serial.begin(115200); // Open serial monitor at 115200 baud to see ping results.

 pingTimer = millis(); // Start now.

servo\_cabeca.attach(11);

servo\_cabeca.write(78);

dir\_zero = 90;

dir\_max\_antihorario = 180;

dir\_max\_horario = 0;

esq\_zero = 90;

esq\_max\_antihorario = 0;

esq\_max\_horario = 180;

}

void loop() {

 // Notice how there's no delays in this sketch to allow you to do other processing in-line while doing distance pings.

 if (millis() >= pingTimer) { // pingSpeed milliseconds since last ping, do another ping.

 pingTimer += pingSpeed; // Set the next ping time.

 sonar.ping\_timer(echoCheck); // Send out the ping, calls "echoCheck" function every 24uS where you can check the ping status.

 }

 // Do other stuff here, really. Think of it as multi-tasking.

int distanceR = 0;

int distanceL = 0;

delay(40);

 if(Distance <= triggerDistance)

 {

 ParadaTotal();

 delay(1000);

 distanceR = OlhaDireita();

 distanceL = OlhaEsquerda();

 servo\_cabeca.write(78);

 if(distanceR>=distanceL)

 {

 VirarDireita();

 delay(TempoGiro);

 }else

 {

 VirarEsquerda();

 delay(TempoGiro);

 }

 }else

 {

 AndarFrente();

 }

 }

void echoCheck() { // Timer2 interrupt calls this function every 24uS where you can check the ping status.

 // Don't do anything here!

 if (sonar.check\_timer()) { // This is how you check to see if the ping was received.

 // Here's where you can add code.

 Distance = sonar.ping\_result / US\_ROUNDTRIP\_CM;

 if(Distance == 0){ //If no ping was recieved

 Distance = MAX\_DISTANCE; //Set the distance to max

 }

 Serial.print("Ping: ");

 Serial.print(Distance); // Ping returned, uS result in ping\_result, convert to cm with US\_ROUNDTRIP\_CM.

 Serial.println("cm");

 }

 // Don't do anything here!

}

int OlhaDireita()

{

 servo\_cabeca.write(0);

 delay(1000);

 int distD = sonar.ping\_cm();

 delay(1000);

 return distD;

}

int OlhaEsquerda()

{

 servo\_cabeca.write(180);

 delay(1000);

 int distE = sonar.ping\_cm();

 delay(1000);

 return distE;

}

void ParadaTotal()

{

 servo\_esteira\_dir.detach();

 servo\_esteira\_esq.detach();

 digitalWrite(12, LOW);

 digitalWrite(13, LOW);

}

void VirarDireita()

{

 servo\_esteira\_dir.attach(12);

 servo\_esteira\_esq.attach(13);

 servo\_esteira\_dir.write(dir\_max\_antihorario);

 servo\_esteira\_esq.write(esq\_max\_horario);

}

void VirarEsquerda()

{

 servo\_esteira\_dir.attach(12);

 servo\_esteira\_esq.attach(13);

 servo\_esteira\_dir.write(dir\_max\_horario);

 servo\_esteira\_esq.write(esq\_max\_antihorario);

}

void AndarFrente()

{

 servo\_esteira\_dir.attach(12);

 servo\_esteira\_esq.attach(13);

 servo\_esteira\_dir.write(dir\_max\_horario);

 servo\_esteira\_esq.write(esq\_max\_horario);

}

void AndarTras()

{

 servo\_esteira\_dir.attach(12);

 servo\_esteira\_esq.attach(13);

 servo\_esteira\_dir.write(dir\_max\_antihorario);

 servo\_esteira\_esq.write(esq\_max\_antihorario);

}