//nano walking robot with ultrasonic sensor

#define BACK\_LEGS\_PIN 10

#define BACK\_LEGS\_GND 8

#define BACK\_LEGS\_VCC 9

#define FRONT\_LEGS\_PIN 3 //front legs on Pin3

#define FRONT\_LEGS\_VCC 2

#define US\_GND 19 //Ultrasonic Ground Pin D19=A5

#define ECHO\_PIN 18 //Ultrasonic Echo Pin D18 = A4

#define TRIGGER\_PIN 17 //Ultrasonic Trigger Pin D17 = A3

#define US\_VCC 16 //Ultrasonic VCC Pin D16 = A2

#define MAXIMUM\_DISTANCE 200

#define SERVO\_BACK\_DISTANCE 60

#define SERVO\_FORWARD\_DISTANCE 100

#define SERVO\_CENTRED 80

#define STEP\_DELAY 160

#define STOP\_DISTANCE 10 //stop distance in cm

#include <Servo.h>

#include <NewPing.h>

//format: front leg position, back leg position

char walkingForward[] = {SERVO\_BACK\_DISTANCE, SERVO\_FORWARD\_DISTANCE,

 SERVO\_FORWARD\_DISTANCE, SERVO\_FORWARD\_DISTANCE,

 SERVO\_FORWARD\_DISTANCE, SERVO\_BACK\_DISTANCE,

 SERVO\_BACK\_DISTANCE, SERVO\_BACK\_DISTANCE

 };

NewPing sonarEyes(TRIGGER\_PIN, ECHO\_PIN, MAXIMUM\_DISTANCE);

Servo servoBackLegs, servoFrontLegs;

void setup() {

 pinMode(LED\_BUILTIN, OUTPUT);

 digitalWrite(LED\_BUILTIN, HIGH);

 //setup Pins to connect Servos To

 pinMode(BACK\_LEGS\_GND, OUTPUT); //tie pin 8 to GND

 digitalWrite(BACK\_LEGS\_GND, LOW);

 pinMode(BACK\_LEGS\_VCC, OUTPUT); //tie pin 9 to VCC

 digitalWrite(BACK\_LEGS\_VCC,HIGH);

 servoBackLegs.attach(BACK\_LEGS\_PIN);

 pinMode(FRONT\_LEGS\_VCC, OUTPUT); //tie pin 2 to VCC

 digitalWrite(FRONT\_LEGS\_VCC, HIGH);

 servoFrontLegs.attach(FRONT\_LEGS\_PIN);

 servoBackLegs.write(SERVO\_CENTRED);

 servoFrontLegs.write(SERVO\_CENTRED);

 //setup Pins for Ultrasonic Sensor

 pinMode(US\_GND, OUTPUT);

 digitalWrite(US\_GND, LOW);

 pinMode(US\_VCC, OUTPUT);

 digitalWrite(US\_VCC,HIGH);

 delay(2000);

 digitalWrite(LED\_BUILTIN, LOW); //turn off built in LED

}

void loop() {

 // put your main code here, to run repeatedly:

 while (checkForObstruction()) { //returns true if obstruction, otherwise skip this loop

 digitalWrite(LED\_BUILTIN, HIGH); //turn on LED for visual indication of obstruction

 //step back and turn

 delay(2000);

 walkBackAndTurnLeft();

 digitalWrite(LED\_BUILTIN, LOW); //turn back off again

 }

 stepForward();

}

void stepForward() {

 for (int n = 0; n < 4; n++) {

 servoFrontLegs.write(walkingForward[n \* 2]);

 servoBackLegs.write(walkingForward[(n \* 2) + 1]);

 delay(STEP\_DELAY);

 }

}

void walkBackAndTurnLeft() {

 for (int n = 0; n < 14; n++) {

 servoFrontLegs.write(SERVO\_CENTRED);

 servoBackLegs.write(SERVO\_BACK\_DISTANCE - 40);

 delay(200);

 servoFrontLegs.write(SERVO\_FORWARD\_DISTANCE);

 servoBackLegs.write(SERVO\_FORWARD\_DISTANCE + 20);

 delay(300);

 }

 servoFrontLegs.write(SERVO\_CENTRED);

 servoBackLegs.write(SERVO\_CENTRED);

 delay(300);

}

bool checkForObstruction() {

 int distance = sonarEyes.ping\_cm();

 if (distance <= STOP\_DISTANCE && distance != 0) {

 return true;

 } else {

 return false;

 }

}