<https://www.youtube.com/watch?v=jQwYBc7YBXs>

#include <AFMotor.h>

#include <Servo.h>

#include <NewPing.h>

#define TRIG\_PIN A0

#define ECHO\_PIN A1

#define MAX\_DISTANCE\_POSSIBLE 100

#define MAX\_SPEED 160

#define MOTORS\_CALIBRATION\_OFFSET 3

#define COLL\_DIST 20

#define TURN\_DIST COLL\_DIST+10

NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE\_POSSIBLE);

AF\_DCMotor leftMotor(1, MOTOR12\_8KHZ);

AF\_DCMotor rightMotor(2, MOTOR12\_8KHZ);

Servo neckControllerServoMotor;

int pos = 0;

int maxDist = 0;

int maxAngle = 0;

int maxRight = 0;

int maxLeft = 0;

int maxFront = 0;

int course = 0;

int curDist = 0;

String motorSet = "";

int speedSet = 0;

void setup() {

  neckControllerServoMotor.attach(10);

  neckControllerServoMotor.write(90);

  delay(2000);

  checkPath();

  motorSet = "FORWARD";

  neckControllerServoMotor.write(90);

  moveForward();

}

void loop() {

  checkForward();

  checkPath();

}

void checkPath() {

  int curLeft = 0;

  int curFront = 0;

  int curRight = 0;

  int curDist = 0;

  neckControllerServoMotor.write(144);

  delay(120);

  for(pos = 144; pos >= 36; pos-=18)

  {

    neckControllerServoMotor.write(pos);

    delay(90);

    checkForward();

    curDist = readPing();

    if (curDist < COLL\_DIST) {

      checkCourse();

      break;

    }

    if (curDist < TURN\_DIST) {

      changePath();

    }

    if (curDist > curDist) {maxAngle = pos;}

    if (pos > 90 && curDist > curLeft) { curLeft = curDist;}

    if (pos == 90 && curDist > curFront) {curFront = curDist;}

    if (pos < 90 && curDist > curRight) {curRight = curDist;}

  }

  maxLeft = curLeft;

  maxRight = curRight;

  maxFront = curFront;

}

void setCourse() {

    if (maxAngle < 90) {turnRight();}

    if (maxAngle > 90) {turnLeft();}

    maxLeft = 0;

    maxRight = 0;

    maxFront = 0;

}

void checkCourse() {

  moveBackward();

  delay(500);

  moveStop();

  setCourse();

}

void changePath() {

  if (pos < 90) {lookLeft();}

  if (pos > 90) {lookRight();}

}

int readPing() {

  delay(70);

  unsigned int uS = sonar.ping();

  int cm = uS/US\_ROUNDTRIP\_CM;

  return cm;

}

void checkForward() { if (motorSet=="FORWARD") {leftMotor.run(FORWARD); rightMotor.run(FORWARD); } }

void checkBackward() { if (motorSet=="BACKWARD") {leftMotor.run(BACKWARD); rightMotor.run(BACKWARD); } }

void moveStop() {leftMotor.run(RELEASE); rightMotor.run(RELEASE);}

void moveForward() {

    motorSet = "FORWARD";

    leftMotor.run(FORWARD);

    rightMotor.run(FORWARD);

  for (speedSet = 0; speedSet < MAX\_SPEED; speedSet +=2)

  {

    leftMotor.setSpeed(speedSet+MOTORS\_CALIBRATION\_OFFSET);

    rightMotor.setSpeed(speedSet);

    delay(5);

  }

}

void moveBackward() {

    motorSet = "BACKWARD";

    leftMotor.run(BACKWARD);

    rightMotor.run(BACKWARD);

  for (speedSet = 0; speedSet < MAX\_SPEED; speedSet +=2)

  {

    leftMotor.setSpeed(speedSet+MOTORS\_CALIBRATION\_OFFSET);

    rightMotor.setSpeed(speedSet);

    delay(5);

  }

}

void turnRight() {

  motorSet = "RIGHT";

  leftMotor.run(FORWARD);

  rightMotor.run(BACKWARD);

  delay(400);

  motorSet = "FORWARD";

  leftMotor.run(FORWARD);

  rightMotor.run(FORWARD);

}

void turnLeft() {

  motorSet = "LEFT";

  leftMotor.run(BACKWARD);

  rightMotor.run(FORWARD);

  delay(400);

  motorSet = "FORWARD";

  leftMotor.run(FORWARD);

  rightMotor.run(FORWARD);

}

void lookRight() {rightMotor.run(BACKWARD); delay(400); rightMotor.run(FORWARD);}

void lookLeft() {leftMotor.run(BACKWARD); delay(400); leftMotor.run(FORWARD);}