#include <SR04.h>

#define TRIG\_PIN 9

#define ECHO\_PIN 10

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Digital Pins

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

const int firstLED = 14, secondLED = 15, thirdLED = 16, fourthLED = 17, fifthLED = 18;

const int buzzer = 19, button = 13;

const int backLeft = 3, backRight = 4, frontLeft = 5, frontRight = 6;

const int forwards = 7, backwards = 8;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Defining Sensors

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SR04 ultraSonic = SR04(ECHO\_PIN, TRIG\_PIN);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Defining Variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

double distance;

boolean pressed = false;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Defining Methods

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void objectDetected();

void startWheels();

void stopWheels();

void rangeHandle();

void rangeHelper(int ledCase);

void turnOffLED();

void buttonWait();

void setup()

{

pinMode(TRIG\_PIN, OUTPUT); // Sets the trigPin as an Output

pinMode(ECHO\_PIN, INPUT); // Sets the echoPin as an Input

for (int index = 3; index < 9; index++)

{

pinMode(index, OUTPUT);

}

for (int index = 14; index < 20; index++)

{

pinMode(index, OUTPUT);

}

pinMode(button, INPUT);

digitalWrite(forwards, HIGH);

digitalWrite(backwards, LOW);

}

void loop()

{

if (!pressed)

{

buttonWait();

}

while (pressed)

{

startWheels();

distance = ultraSonic.Distance(); // Distance in cm

rangeHandle();

if (distance <= 45 && distance > 2)

{

objectDetected();

}

}

}

void objectDetected()

{

stopWheels();

digitalWrite(forwards, LOW);

digitalWrite(backwards, HIGH);

startWheels();

delay(1500);

stopWheels();

while (distance <= 45 && distance > 2)

{

digitalWrite(backRight, HIGH);

digitalWrite(frontRight, HIGH);

delay(3000);

stopWheels();

delay(2000);

distance = ultraSonic.Distance();

}

digitalWrite(forwards, HIGH);

digitalWrite(backwards, LOW);

}

void startWheels()

{

digitalWrite(frontRight, HIGH);

digitalWrite(frontLeft, HIGH);

digitalWrite(backRight, HIGH);

digitalWrite(backLeft, HIGH);

}

void stopWheels()

{

digitalWrite(frontRight, LOW);

digitalWrite(frontLeft, LOW);

digitalWrite(backRight, LOW);

digitalWrite(backLeft, LOW);

}

void rangeHandle()

{

// 2cm to 400cm

if (distance < 2 || distance > 350)

{

rangeHelper(0);

}

else if (distance > 320 && distance <= 400)

{

rangeHelper(1);

}

else if (distance > 240 && distance <= 320)

{

rangeHelper(2);

}

else if (distance > 160 && distance <= 240)

{

rangeHelper(3);

}

else if (distance > 80 && distance <= 160)

{

rangeHelper(4);

}

else if (distance <= 80 && distance > 2)

{

rangeHelper(5);

}

}

void rangeHelper(int ledCase)

{

turnOffLED();

switch (ledCase)

{

case 0:

turnOffLED();

noTone(buzzer);

case 1:

digitalWrite(firstLED, HIGH);

tone(buzzer, 100);

break;

case 2:

digitalWrite(firstLED, HIGH);

digitalWrite(secondLED, HIGH);

tone(buzzer, 200);

break;

case 3:

digitalWrite(firstLED, HIGH);

digitalWrite(secondLED, HIGH);

digitalWrite(thirdLED, HIGH);

tone(buzzer, 400);

break;

case 4:

digitalWrite(firstLED, HIGH);

digitalWrite(secondLED, HIGH);

digitalWrite(thirdLED, HIGH);

digitalWrite(fourthLED, HIGH);

tone(buzzer, 600);

break;

case 5:

digitalWrite(firstLED, HIGH);

digitalWrite(secondLED, HIGH);

digitalWrite(thirdLED, HIGH);

digitalWrite(fourthLED, HIGH);

digitalWrite(fifthLED, HIGH);

tone(buzzer, 800);

break;

}

}

void turnOffLED()

{

digitalWrite(firstLED, 0);

digitalWrite(secondLED, 0);

digitalWrite(thirdLED, 0);

digitalWrite(fourthLED, 0);

digitalWrite(fifthLED, 0);

}

void buttonWait()

{

int buttonState = 0;

while (true)

{

buttonState = digitalRead(button);

if (buttonState == HIGH)

{

delay(500);

buttonState = digitalRead(button);

if (buttonState == HIGH)

{

pressed = true;

return;

}

}

}

}