#include <LiquidCrystal.h>

#include<Blynk.h>

#include <BlynkSimpleStream.h>

//Your app authentication token (can be fetched from your blynk app.

char auth[] = "XXXXXXXXXXXXXXXXXXXXXXXXXX";//Enter blynk auth token

BlynkTimer timer;

LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

const int lm1 = 2;

const int lm2 = 3;

const int rm1 = 4;

const int rm2 = 5;

long duration, inches, cm;

const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor

const int buzzer=A0;

const int echoPin = 6; // Echo Pin of Ultrasonic Sensor

void sendSensor()

{

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(10);

digitalWrite(pingPin, LOW);

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

if(cm < 390)

{

buzzer\_func();

stop();

delay(100);

go\_back();

delay(100);

stop\_again();

delay(100);

go\_right();

delay(100);

}

else

{

go\_straight();

delay(100);

}

Blynk.virtualWrite(V5, cm);

if(cm<380)

{

Blynk.email("ananthavadthani@gmail.com", "OBSTACLE ALERT!", "FALL DETECTED");

}

}

void setup() {

// put your setup code here, to run once:

pinMode(lm1, OUTPUT);

pinMode(lm2, OUTPUT);

pinMode(rm1, OUTPUT);

pinMode(rm2, OUTPUT);

pinMode(buzzer,OUTPUT);

lcd.begin(16, 2);

Serial.begin(9600);

Blynk.begin(auth, Serial);

timer.setInterval(2000L, sendSensor);

}

void loop() {

Blynk.run();

timer.run();

Blynk.notify("Device started");

}

long microsecondsToInches(long microseconds)

{

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds)

{

return microseconds / 29 / 2;

}

void go\_straight()

{

lcd.setCursor(0,0);

lcd.print("NOTHING AHEAD");

lcd.setCursor(0,1);

lcd.print("SAFE TO MOVE FORWARD");

digitalWrite(lm1,HIGH);

digitalWrite(lm2,LOW);

digitalWrite(rm1,HIGH);

digitalWrite(rm2,LOW);

}

void go\_back()

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("TAKING REVERSE");

lcd.setCursor(0,1);

lcd.print(cm);

digitalWrite(lm2,HIGH);

digitalWrite(lm1,LOW);

digitalWrite(rm2,HIGH);

digitalWrite(rm1,LOW);

}

void stop()

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SOMETHING AHEAD");

lcd.setCursor(0,1);

lcd.print("STOP!!");

digitalWrite(lm1,LOW);

digitalWrite(lm2,LOW);

digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

}

void stop\_again()

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("STOP FOR TURN");

digitalWrite(lm1,LOW);

digitalWrite(lm2,LOW);

digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

}

void go\_right()

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("TURNING RIGHT");

lcd.setCursor(0,1);

lcd.print(cm);

digitalWrite(lm1,HIGH);

digitalWrite(lm2,LOW);

digitalWrite(rm1,LOW);

digitalWrite(rm2,LOW);

}

void buzzer\_func()

{

tone(buzzer,1000);

delay(200);

noTone(buzzer);

}