#include <TinyGPS++.h>

#include <Wire.h>

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

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

#define r 6

#define alm 5

#define prox A2

#define prox2 A3

#define sw 4

#define u1\_triger\_high digitalWrite(A0, HIGH)

#define u1\_triger\_low digitalWrite(A0, LOW)

#define ultra1 A1

unsigned char count;

unsigned a = 0;

int sec = 0, secc = 0, yy,ult;

char g, aq = 0;

char lat[25], lon[25];

static const int RXPin = 2, TXPin = 3;

static const uint32\_t GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial ss(RXPin, TXPin);

#define REPORTING\_PERIOD\_MS 1000

uint32\_t tsLastReport = 0;

unsigned long previousMillis = 0;

const long interval = 1000;

void setup() {

  pinMode(prox, INPUT);

  pinMode(prox2, INPUT);

  pinMode(r, OUTPUT);

  pinMode(alm, OUTPUT);

  digitalWrite(r, LOW);

  pinMode(sw, INPUT\_PULLUP);

  pinMode(A0, OUTPUT);

  pinMode(A1, INPUT);

  Serial.begin(9600);

  ss.begin(GPSBaud);

  lcd.begin(16, 2);

  lcd.setCursor(0, 0);

  lcd.print("----------------");

  lcd.setCursor(0, 1);

  lcd.print("----------------");

  delay(2000);

  lcd.clear();

}

void loop() {

  if (digitalRead(sw) == LOW) {

    aq = 1;

  } else {

    aq = 0;

  }

  while (ss.available() > 0) {

    gps.encode(ss.read());

    if (gps.location.isUpdated()) {

      // GPS data updated

    }

  }

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    sec++, secc++;

  }

  // Check ultrasonic distance

   ult = read\_ultrasonic1();

  lcd.setCursor(0, 0);

  lcd.print("DIS:");

  Lcd\_Decimal3(4, 0, ult);

  // Proximity or ultrasonic object detection

  if ((digitalRead(prox) == HIGH) || (digitalRead(prox2) == HIGH)) {

    a = 3;

    digitalWrite(r, LOW);

    digitalWrite(alm, HIGH);

    lcd.setCursor(0, 1);

    lcd.print(" CRACK\_DETECTED ");

  } else if (ult < 15) {

    a = 4;

    digitalWrite(r, LOW);

    digitalWrite(alm, HIGH);

    lcd.setCursor(0, 1);

    lcd.print(" OBJECT\_DETECTED");

  } else {

    a = 0;

    digitalWrite(r, HIGH);

    digitalWrite(alm, LOW);

    lcd.setCursor(0, 1);

    lcd.print("     NORMAL     ");

  }

  if (sec > 5) {

    http\_send();

    sec = 0;

  }

}

void http\_send() {

  Serial.print("send to iot");

  if (aq == 1) {

    Serial.print(gps.location.lat(), 6);

    Serial.print(",");

    Serial.print(gps.location.lng(), 6);

  } else {

    Serial.print("11.65753717392018,78.2603105187629");

  }

  Serial.print(",");

  if (a == 3) {

    Serial.print("CRACK\_DETECTED");

  } else if (a == 4) {

    Serial.print("OBJECT\_DETECTED");

  } else {

    Serial.print("NORMAL");

  }

  Serial.write(0x0d);  // carriage return

  Serial.write(0x0a);  // newline

}

unsigned int read\_ultrasonic1() {

  int ultrasonic = 0;

  u1\_triger\_low;

  delay(1);

  u1\_triger\_high;

  delay(10);

  u1\_triger\_low;

  ultrasonic = pulseIn(ultra1, HIGH) / 56.0;

  return ultrasonic;

}

void Lcd\_Decimal3(unsigned char com, unsigned char com1, unsigned int val) {

  unsigned int Lcd\_h, Lcd\_hr, Lcd\_t, Lcd\_o;

  lcd.setCursor(com, com1);

  Lcd\_h = val / 100;

  Lcd\_hr = val % 100;

  Lcd\_t = Lcd\_hr / 10;

  Lcd\_o = Lcd\_hr % 10;

  lcd.setCursor(com, com1);

  lcd.write(Lcd\_h + 0x30);

  lcd.setCursor(com + 1, com1);

  lcd.write(Lcd\_t + 0x30);

  lcd.setCursor(com + 2, com1);

  lcd.write(Lcd\_o + 0x30);

}