**Obstacle Avoidance System**

**📘 Project Title:**

**Smart Obstacle Avoidance System with LCD & Blynk Cloud Status**

**📋 Project Description:**

This project detects obstacles using an IR sensor and responds by displaying messages on an LCD and Blynk dashboard. It also uses colored LEDs and a buzzer to indicate whether the path is clear or blocked.

**🔩 Components Used**

| **Component Name** | **Pin No** | **Destination Component** | **Pin No** | **Special Remark** |
| --- | --- | --- | --- | --- |
| IR Sensor (Digital) | OUT | ESP32 | D33 | Detects obstacle (LOW if detected) |
| Red LED | + | ESP32 | D26 | Indicates obstacle |
| Green LED | + | ESP32 | D25 | Indicates clear path |
| Buzzer | + | ESP32 | D27 | Audible alert for obstacle |
| I2C LCD (16x2) | SDA | ESP32 | D21 | I2C Data Line |
| I2C LCD (16x2) | SCL | ESP32 | D22 | I2C Clock Line |
| All VCCs | VCC | ESP32 | 3.3V/5V | Check sensor input voltage range |
| All Grounds | GND | ESP32 | GND | Common Ground |

**📦 Required Libraries**

Install these in **Arduino IDE > Library Manager** (Ctrl + Shift + I or Tools > Manage Libraries):

| **Library Name** | **Description** |
| --- | --- |
| **Blynk** (v1.1.0 or newer) | For Blynk IoT cloud communication |
| **LiquidCrystal\_I2C** | For I2C LCD display |
| **Wire** (built-in) | Enables I2C communication |

**🧠 Circuit Connections**

**(URL:-https://app.cirkitdesigner.com/project/dc5a8b86-2013-491c-b7ad-1c39681dec48)**

**🧠 Project Logic Summary**

| **Condition** | **Output on LCD** | **Output on Blynk** | **LED/Buzzer Action** |
| --- | --- | --- | --- |
| Obstacle detected (LOW) | "Obstacle Detected" | Same | Red LED ON, Buzzer buzzes |
| No obstacle (HIGH) | "Clear Path! Proceed" | Same | Green LED ON, buzzer OFF |

**🧾 Arduino Code with Explanation**

// ----------------------------

// BLYNK CREDENTIALS (From Template)

// ----------------------------

#define BLYNK\_AUTH\_TOKEN "nTP7y3tPvDjYw2JG\_GcIkxIL6YADGYHn"

#define BLYNK\_TEMPLATE\_ID "TMPL3klgoZxWl"

#define BLYNK\_TEMPLATE\_NAME "ondyicaleAvoidance"

// ----------------------------

// Include Required Libraries

// ----------------------------

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

// ----------------------------

// Define Component Pins

// ----------------------------

#define IR\_SENSOR\_PIN 33

#define BUZZER\_PIN 27

#define RED\_LED 26

#define GREEN\_LED 25

// ----------------------------

// LCD Object (I2C Address 0x27, 16x2)

// ----------------------------

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// ----------------------------

// WiFi Credentials

// ----------------------------

char ssid[] = "YourWiFiSSID"; // ← Replace with your WiFi name

char pass[] = "YourWiFiPassword"; // ← Replace with your WiFi password

void setup() {

Serial.begin(115200);

// Setup GPIO Modes

pinMode(IR\_SENSOR\_PIN, INPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(RED\_LED, OUTPUT);

pinMode(GREEN\_LED, OUTPUT);

// LCD Setup

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Starting System");

delay(1500);

lcd.clear();

// Connect to WiFi + Blynk Cloud

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssid, pass);

}

void loop() {

Blynk.run();

int irValue = digitalRead(IR\_SENSOR\_PIN);

if (irValue == LOW) { // Obstacle Detected

lcd.setCursor(0, 0);

lcd.print("Obstacle Detected");

lcd.setCursor(0, 1);

lcd.print("Avoid this path ");

digitalWrite(RED\_LED, HIGH);

digitalWrite(GREEN\_LED, LOW);

digitalWrite(BUZZER\_PIN, HIGH);

delay(300);

digitalWrite(BUZZER\_PIN, LOW);

delay(300);

Blynk.virtualWrite(V0, "Obstacle Detected! Avoid this path");

} else { // Clear Path

lcd.setCursor(0, 0);

lcd.print("Clear Path! ");

lcd.setCursor(0, 1);

lcd.print("Proceed Safely ");

digitalWrite(RED\_LED, LOW);

digitalWrite(GREEN\_LED, HIGH);

digitalWrite(BUZZER\_PIN, LOW);

Blynk.virtualWrite(V0, "Clear Path! Proceed");

delay(500);

}

}

**🌐 Blynk IoT Dashboard Setup**

**Step 1: Create Account**

* Go to [https://blynk.cloud](https://blynk.cloud/)
* Create a new account and login

**Step 2: Create Template**

* Click “**+ New Template**”
* Name: ondyicaleAvoidance
* Hardware: ESP32
* Connection: WiFi
* Save Template

**Step 3: Add Datastream**

* Type: **Virtual Pin**
* Pin: V0
* Name: ObstacleStatus
* Data Type: **String**
* Save

**Step 4: Create Device**

* From Template: ondyicaleAvoidance
* Name: anything (e.g. ObstacleBot)
* Copy Auth Token → Paste into Arduino sketch

**Step 5: Web Dashboard**

* Go to Device > Web Dashboard
* Drag **Label** widget
* Bind it to Datastream **V0**
* Name it: "Obstacle Message"

**Step 6: (Optional) Mobile App**

* Install **Blynk IoT** app on your phone
* Login with same credentials
* View live status of obstacles!