# **IR Sensor Web Dashboard**

# Overview:

This project monitors an **IR sensor** using a **Jetson Nano**, provides real-time status updates via a **Flask web server**, and displays a **live dashboard** with:  
✅ **Object Detection Alerts**  
✅ **Detection Counter**  
✅ **Detection History** (Last 5 events)

Required Components:

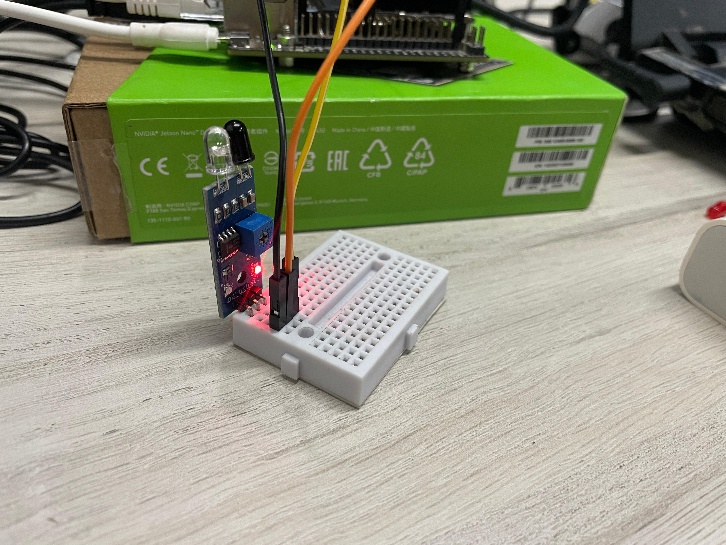
* Jetson Nano
* IR Sensor Module (Active Low)
* Jumper Wires

Software Dependencies

* Python 3
* Flask (for web server)
* Jetson.GPIO (for GPIO control)

## **Circuit Connections :**

|  |  |
| --- | --- |
| **IR Sensor Pin** | **Jetson Nano Pin** |
| |  | | --- | | VCC (+) |  |  | | --- | |  | | 3.3V / 5V |
| |  | | --- | | GND (-) |  |  | | --- | |  | | GND |
| OUT | GPIO 18 (BOARD Mode) |



# Install Required Packages:

pip3 install Jetson.GPIO

pip3 install Jetson.GPIO

# Code Implementation:

**main.py (Backend - Flask & Jetson GPIO)**

**This script:  
✅ Reads the IR sensor status.  
✅ Updates a live counter for object detections.  
✅ Stores the last 5 detection events.  
✅ Runs a Flask web server for real-time monitoring.**

CODE:

import Jetson.GPIO as GPIO

import time

from flask import Flask, render\_template, jsonify

import threading

# GPIO Setup

IR\_SENSOR\_PIN = 18

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(IR\_SENSOR\_PIN, GPIO.IN)

# Flask Web App

app = Flask(\_\_name\_\_)

# Global variables

sensor\_value = 1 # Default (No object)

detection\_count = 0 # Count the number of detections

detection\_history = [] # Store last 5 detections

# Function to monitor the IR sensor

def monitor\_sensor():

global sensor\_value, detection\_count, detection\_history

while True:

new\_value = GPIO.input(IR\_SENSOR\_PIN) # Read sensor value

if new\_value == 0 and sensor\_value == 1: # Object detected (falling edge)

detection\_count += 1

detection\_history.append(time.strftime("%H:%M:%S")) # Add timestamp

if len(detection\_history) > 5: # Keep only last 5 records

detection\_history.pop(0)

sensor\_value = new\_value # Update the global value

time.sleep(0.5) # Update every 500ms

# Routes

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/sensor\_data")

def sensor\_data():

return jsonify({

"sensor\_value": sensor\_value,

"detection\_count": detection\_count,

"detection\_history": detection\_history

})

# Start background sensor monitoring

threading.Thread(target=monitor\_sensor, daemon=True).start()

# Run Flask app

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="0.0.0.0", port=5000, debug=True)

**Templates/index.html (Frontend - Web Dashboard)**

**This web interface:  
✅ Shows real-time sensor status.  
✅ Displays a detection counter.  
✅ Lists the last 5 detection events.  
✅ Includes a dark mode toggle for UI customization.**

**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>IR Sensor Dashboard</title>

<style>

body { font-family: Arial, sans-serif; text-align: center; padding: 20px; transition: 0.3s; }

.status-box { font-size: 24px; padding: 20px; border-radius: 10px; display: inline-block; }

.safe { background: green; color: white; }

.danger { background: red; color: white; animation: blink 0.5s infinite alternate; }

@keyframes blink { from { opacity: 1; } to { opacity: 0.5; } }

.dark-mode { background: #333; color: white; }

button { padding: 10px; margin: 10px; cursor: pointer; }

</style>

<script>

function updateStatus() {

fetch('/sensor\_data')

.then(response => response.json())

.then(data => {

let statusBox = document.getElementById("sensorStatus");

let counterBox = document.getElementById("detectionCount");

let historyList = document.getElementById("detectionHistory");

// Update status

if (data.sensor\_value === 0) {

statusBox.innerText = "⚠ Object Detected!";

statusBox.className = "status-box danger";

} else {

statusBox.innerText = "✔ No Object";

statusBox.className = "status-box safe";

}

// Update counter

counterBox.innerText = "Detections: " + data.detection\_count;

// Update history

historyList.innerHTML = "";

data.detection\_history.forEach(time => {

let li = document.createElement("li");

li.textContent = "Detected at " + time;

historyList.appendChild(li);

});

});

}

function toggleDarkMode() {

document.body.classList.toggle("dark-mode");

}

setInterval(updateStatus, 500);

</script>

</head>

<body>

<h1>IR Sensor Monitoring</h1>

<button onclick="toggleDarkMode()">🌙 Toggle Dark Mode</button>

<div id="sensorStatus" class="status-box safe">✔ No Object</div>

<p id="detectionCount">Detections: 0</p>

<h3>Detection History</h3>

<ul id="detectionHistory"></ul>

</body>

</html>

**Running the Project**

Follow these steps to run the project:

**🟢 Step 1: Setup the Environment**

Ensure you are in the project directory and all dependencies are installed.

**🟢 Step 2: Start the Flask Web Server**

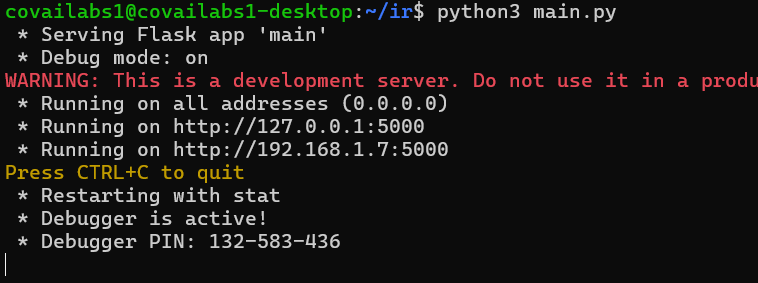
Run the Python script:

python3 main.py

You'll see an output like:

\* Running on http://127.0.0.1:5000

\* Running on <http://192.168.1.X:5000>



**🟢 Step 3: Access the Web Dashboard**

Open a browser and go to:

http://192.168.1.X:5000

