International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

DISASTER RESCUE HUMAN DETECTION ROBOT

B. Phanindar Kumar[1], M.Akhil [2], G. Pavankumar[3], N. Sravan [4*]

[1] Assistant Professor Mechanical Engineering ,Guru Nanak Institute Of Technology ,Hyderabad ,Telangana
[2] UG scholars , Mechanical Engineering ,Guru Nanak Institute Of Technology ,Hyderabad ,Telangana
[3,4] UG scholars , Mechanical Engineering ,Guru Nanak Institute Of Technology ,Hyderabad ,Telangana

ABSTRACT :

Disasters often create life-threatening situations where quick and efficient rescue operations are critical. The Disaster Rescue Human Detection Robot is a compact, efficient robotic system designed to locate humans in disaster-struck areas using a combination of hardware and software technologies. This robot is powered by a NodeMCU ESP8266 microcontroller, which integrates seamlessly with a Wi-Fi module for real-time communication. A Passive Infrared (PIR) sensor is employed to detect human presence by sensing body heat and motion.

The robot is equipped with a lithium-ion battery for efficient power supply and uses an L298N motor driver to control its five DC motors, enabling smooth navigation across challenging terrains. A power distribution board ensures stable operation of all components. Additionally, a buzzer serves as an alert system, indicating the detection of humans.

The integration of the ESP8266 microcontroller with Arduino and Wi-Fi modules allows the robot to transmit live data to rescue teams, facilitating timely decision-making. Its compact design and autonomous capabilities make it a valuable asset for locating survivors in confined and dangerous environments. This robotic system not only enhances the safety of rescue operations but also accelerates the process of locating and saving lives, showcasing a significant advancement in disaster management technology.

Keywords: Human Detection, Disaster Rescue, PIR Sensor, ESP8266 Microcontroller, Autonomous Navigation, Wi-Fi Module Integration.

1. Introduction :

Disasters such as earthquakes, building collapses, and other emergencies often leave victims trapped in hazardous and inaccessible areas. Effective search and rescue operations require advanced tools to detect and locate survivors quickly and efficiently. The Disaster Rescue Human Detection Robot is designed to address these challenges by leveraging modern robotic and sensor technologies. This robot utilizes a PIR (Passive Infrared) sensor to detect human presence by sensing body heat and movement, ensuring accurate identification even in low-visibility conditions. The core of the system is the ESP8266 microcontroller, which facilitates communication and control by interfacing with Arduino and a Wi-Fi module. This enables real-time data transmission and remote monitoring. Powered by a lithium-ion battery, the robot features a NodeMCU ESP8266 power distribution board to manage energy flow efficiently. The L298N motor driver controls five DC motors, providing mobility and stability on uneven terrain. A buzzer serves as an alert system, signaling human detection to rescue teams. Compact, reliable, and easy to deploy, the Disaster Rescue Human Detection Robot enhances the speed and safety of search operations. By combining advanced sensors, autonomous movement, and wireless communication, this robot offers a practical and innovative solution for disaster management and rescue missions.

2. Design of Disaster Rescue Human Detection Robot :

2.1. Mechanical Design:

The mechanical structure ensures mobility, durability, and adaptability in disaster-stricken terrains.

- **Chassis**: A compact, lightweight yet sturdy base to house all components. Typically made of aluminium or high-strength plastic.
- **Motors and Wheels**:
  5 DC Motors: Provide the drive system for movement. Four motors for locomotion and one motor for additional mechanisms like a robotic arm (optional).
  Wheels/Tracks: Rubberized wheels or tracked systems enhance grip on rubble or uneven surfaces.
- **Mounting**: Components like the PIR sensor, battery, microcontroller, and buzzer are securely mounted on the chassis.

2.2. Electronics Design:

The electronics setup ensures seamless operation and integration of sensors, controllers, and communication modules**.**

- **NodeMCU ESP8266 Microcontroller**: Acts as the brain of the robot, controlling the sensors, motors, and Wi-Fi communication.
- **Power Supply:**
  - **Lithium-Ion Battery**: Provides power to all components.
  - **Power Distribution Board**: *Distributes power efficiently to prevent overloading and ensures stable operation***.**
- **PIR Sensor**: Mounted at the front, detects body heat to identify humans**.**
- **L298N Motor Driver**: Controls the DC motors, enabling forward, backward, and turning movements**.**
- **Buzzer**: Sounds an alert when the PIR sensor detects a human presence.
- **Wi-Fi Module (Integrated with ESP8266*):*** Allows real-time data communication with a remote monitoring system**.**

2.2. Process Workflow:

- **Initialization**: The microcontroller initializes all components (PIR sensor, motor driver, and communication module) when powered on.
- **Navigation**:
  - The robot uses predefined navigation patterns or manual control (via Wi-Fi) to move through the disaster area.
  - DC motors, controlled by the L298N motor driver, handle movement.
- **Human Detection**: The PIR sensor continuously monitors the surroundings for heat signatures indicative of human presence.
- **Alert Mechanism**: If the PIR sensor detects a human, the microcontroller triggers the buzzer and sends an alert to the rescue team via the Wi-Fi module.
- **Data Transmission**: The ESP8266 sends location data (if GPS is integrated) and detection status to the rescue team's monitoring system or a mobile device.
- **Power Management**: The lithium-ion battery and power distribution board ensure that the robot can operate for extended periods without interruption.

3. Block Diagram of Process:

[Power Supply]
↓
[Power Distribution Board]
↓
[ESP8266 Microcontroller]
↓
[PIR Sensor | L298N Motor Driver | Wi-Fi Module]
↓
[Buzzer | DC Motors | Communication System]

Key Features:

- Autonomous Operation: Minimal human intervention required for navigation and detection.
- Real-Time Communication: Wi-Fi module transmits alerts and status to rescue teams.
- Compact Design: Easily maneuvers through tight or obstructed spaces.
- Durability: Rugged design ensures reliability in harsh environments.

4. Software Used:

The Arduino Integrated Development Environment (IDE) is an open-source software for writing, compiling, and uploading code to Arduino boards. It features a text editor for creating "sketches" (programs), with tools for syntax highlighting, brace matching, and automatic indentation. The IDE includes a message area for feedback, a console for error messages, and a toolbar for common functions like verifying and uploading code. Sketches are stored in a "sketchbook" directory, which can be customized. It supports importing libraries, including those in ZIP format, to extend functionality. The cross-platform software, written in Java, runs on Windows, Mac OS X, and Linux, making it accessible and easy to use for developers and beginners alike.

Fig. 1.1 Arduino IDE logo

Code generated for Robot:

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char *ssid = "Human_Rescue10";
const char *password = "12345678";
int s1=70,s2=90;
ESP8266WebServer server(80);
String page =""; //For the Web Server
String page2="";  //For updating Status of robot
int sensor=10;
int moment=0;
String mstr;
void setup() {
page="<center><h1>Weed Cutting Robot</h1><body><p><a href=\"F\"><button>forward</button></a></p><p><center><a href=\"B\"><button>backward</button></a></p><center><p><a href=\"R\"><button>right</button></a></p><center><p><a href=\"L\"><button>left</button></a></p><p><center><a href=\"S\"><button>Stop</button></a></p><p><a href=\"a\"><button>CUT_ON</button></a></p><p><a href=\"b\"><button>CUT_OFF</button></a></p></body>";
delay(1000);
pinMode(D1,OUTPUT);
pinMode(D2,INPUT);
pinMode(D3,OUTPUT);
pinMode(D4,OUTPUT);
pinMode(D5, OUTPUT);   // inputs for motor 1
pinMode(D6,OUTPUT);
pinMode(D7,OUTPUT);    // inputs for motor 2
pinMode(D8,OUTPUT);
pinMode(LED_BUILTIN,OUTPUT);  // For status of WiFi connection
digitalWrite(D5,LOW);
digitalWrite(D6,LOW);
digitalWrite(D7,LOW);
digitalWrite(D8,LOW);
Serial.begin(115200);
Serial.println();
Serial.print("Configuring access point…");
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(myIP);

server.begin();
Serial.println("HTTP server started");
  server.on("/", webpage);
  server.on("/F",Forward);
  server.on("/B",Backward);
  server.on("/L",Left);
  server.on("/R",Right);
  server.on("/S",stop1);
 // server.on("/a",cut_on);// turns all the motor input pins low
 // server.on("/b",cut_off);
  delay(200);
  server.begin();
  Serial.println("Web server started!");
}

void loop() {
//mstr=String(sensor);
```

```
server.handleClient();
smoke=digitalRead(D2);
Serial.println(moment);
if(moment==1)
{

 //server.handleClient();
 //
  //delay(1000);
  digitalWrite(D3,LOW);
  delay(500);

}
else
{
  digitalWrite(D3,HIGH);
}

//server.send(200,"text/plain",mstr);

}

void webpage()
{
server.send(200, "text/html", page+page2);
}


void Forward()
 {
    //analogWrite(D3,60);
    //analogWrite(D4,80);
    digitalWrite(D5,LOW);
    digitalWrite(D6,HIGH);
    digitalWrite(D7,HIGH);
    digitalWrite(D8,LOW);
    Serial.print('F');
    page2="<center><p> Robot Status : Forward </p></center>";
    server.send(200,"text/html", page+page2);
   // delay(200);
  }
  void Left()
  {
    page2="<center><p> Robot Status : Left</p></center>";
    server.send(200,"text/html",page+page2);
  //  analogWrite(D3,s1);
  //  analogWrite(D4,s2);
    digitalWrite(D5,LOW);
    digitalWrite(D6,LOW);
    digitalWrite(D7,HIGH);
    digitalWrite(D8,LOW);
    delay(200);
    Serial.print('L');
  }
  void Right()
   {
    //analogWrite(D3,s1);
    //analogWrite(D4,s2);
    page2="<center><p> Robot Status : Right</p></center>";
    server.send(200,"text/html",page+page2);
```

```
  digitalWrite(D5,LOW);
  digitalWrite(D6,HIGH);
  digitalWrite(D7,LOW);
  digitalWrite(D8,LOW);
  delay(200);
  Serial.print('R');
 }
 void Backward()
 {
  page2="<center><p> Robot Status : Backward</p></center>";
  server.send(200, "text/html", page+page2);
  //analogWrite(D3,s1);
  //analogWrite(D4,LOW);
  digitalWrite(D5,HIGH);
  digitalWrite(D6,LOW);
  digitalWrite(D7, LOW);
  digitalWrite(D8,HIGH);
  delay(200);
  Serial.print('B');
 }
 void stop1()
 {
   page2="<center><p> Robot Status : Stop</p></center>";
//  page3="<center><p> motor 2 Status : off</p></center>";
 server.send(200,"text/html",page+page2);
 //analogWrite(D3,s1);
  //analogWrite(D4,LOW);
  digitalWrite(D5,LOW);
  digitalWrite(D6,LOW);
  digitalWrite(D7,LOW);
  digitalWrite(D8,LOW);
  Serial.print('S');
 }
/*  void cut_on()
 {
  String page5="<center><p>Weed Cutting</p></center>";
  server.send(200,"text/html",page+page5);
  digitalWrite(D1,HIGH);
  digitalWrite(D2,LOW);
  digitalWrite(D3,HIGH);
  digitalWrite(D4,LOW);
 // digitalWrite(D3,LOW);
//  digitalWrite(D4,HIGH);
 // delay(3000);

 }
 void cut_off()
 {
  String page6="<center><p> Bullet firing stopped</p></center>";
  server.send(200,"text/html",page+page6);
  digitalWrite(D1,LOW);
  digitalWrite(D2,LOW);
  digitalWrite(D3,LOW);
  digitalWrite(D4,LOW);
 }*/
```

Program Result:

- ***Wi-Fi Setup*:**
  - Creates a Wi-Fi hotspot (Human_Rescue10, password: 12345678).
  - Access control page at ESP's IP (e.g., 192.168.4.1).
- ***Web Control:***
  - Buttons for robot movement: Forward, Backward, Left, Right, Stop.
  - Clicking updates robot status on the webpage.
- ***Robot Actions:***
  - Moves as per commands via motor driver.
  - Stops or changes direction based on input.
- ***Sensor Monitoring:***
  - If sensor (D2) is triggered, buzzer (D3) activates briefly.
  - Sensor state printed to Serial Monitor.
- ***Serial Output:***
  - Logs commands and sensor data (e.g., F, moment: 1).
- ***Behaviour:***
  - Physical robot responds to web commands and sensor input.

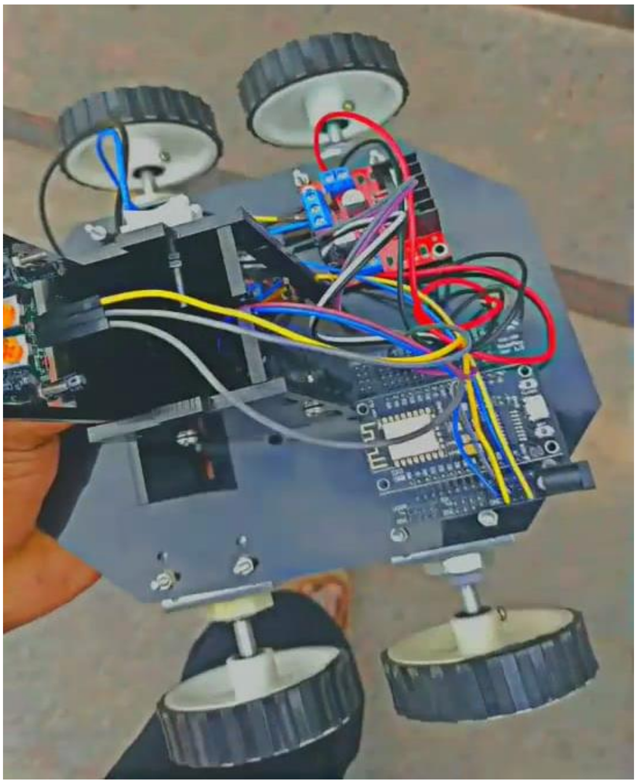

Fig.1.2. Disaster Rescue Human Detection Robot

Advantages of the Disaster Rescue Human Detection Robot:

- ***Enhanced Safety*:**
  - Operates in dangerous environments, reducing risks to human rescuers.
- ***Quick Detection*:**
  - Equipped with a PIR sensor for detecting humans, speeding up rescue operations.
- ***Remote Accessibility:***
  - Controlled wirelessly via a Wi-Fi interface, allowing operation from a safe distance.
- ***Real-time Monitoring:***
  - Sensor data and robot status are updated live, aiding decision-making.
- ***Adaptability:***
  - Can be equipped with additional sensors (e.g., smoke, gas) for multi-functional use in various disaster scenarios.

- ***Compact and Portable:***
  - Small size allows it to navigate through tight spaces in collapsed structures.
- ***Automation Support*:**
  - Reduces manual effort by automating movement and detection processes.
- ***Cost-effective:***
  - Utilizes affordable components like ESP8266, DC motors, and basic sensors, making it budget-friendly for deployment.

9.Acknowledgements:

We sincerely thank our internal guide, Mr. B. Phanindar Kumar, for his valuable advice and encouragement throughout this project. Our gratitude also extends to Dr. B. Vijaya Kumar, Professor and Head of the Mechanical Department, for his expert supervision and insightful suggestions. We appreciate Principal Dr. Koduganti Venkata Rao and the management for providing the necessary facilities. Additionally, we thank the Mechanical Engineering staff and lab technicians for their assistance.

10.REFERENCES:

1. Brooks, Rodney. "Achieving Artificial Intelligence through Building Robots." Boston: Massachusetts Institute of Technology, 1986.
2. Horswill, Ian. "The Polly System." AI and Mobile Robots.
3. Mark Yim, David G. Duff, and Kimon D. Roufas. "PolyBot: a Modular Reconfigurable Robot." IEEE International Conference on Robotics & Automation. April 2000.
4. Vaughan, R. N. Sumpter, A. Frost, and S. Cameron. "Experiments in Automatic Flock Control." Edinburgh, UK, 1998.
5. Martin, Martin C. and Hans Moravec. "Robot Evidence Grids." CMU RI TR 96-06, 1996.
6. Illah Nourbakhsh, Rob Powers, and Stan Birchfield. "DERVISH: An Office-Navigating Robot." Copyright 1995, AAAI.
7. Moravec, Hans. "Robots, After All." Communications of the ACM. October 2003. Vol. 46, No. 10.
8. Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots." Copyright 1999, AAAI.
9. R. Grabowski, L. Navarro-Serment, and P. Khosla. "An Army of Small Robots." SciAm Online May 2004.
10. Thrun, Sebastian. "Robotic Mapping: A Survey." CMU-CS-02-111, February 2002.
11. LaValle, Steven and James Kuffner. "RRT-Connect: An Efficient Approach to Single-Query Path Planning." Copyright 2000, ICRA.