# Rescue robot for natural disaster zones

Student Scientific Communications Session (SCSS-AIA-2022), May 20, 2022 edition

Department of Automation and Applied Informatics

Robert Cătălin Diaconu

Systems Applied in Production and Services

Polytechnic University of Timișoara
Timișoara, Romania
robert.diaconu@student.upt.ro

***Abstract*— This paper describes the architecture and development stages of a remote-controlled robot, with the main purpose of finding victims in natural disaster affected areas. Making use of integrated video camera, robot can capture and send in real time a continuous stream of images with speeds up to 60 FPS. Camera position can be adjusted remotely, acting independent of robot's moves. Most valuable feature is the wireless connection, which facilitates communication with servers, for statistical data, and with the person who is controlling the robot. Although it is constantly monitored, as a protection measure the robot is equipped with proximity sensors, to avoid obstacles and to travel on rough roads. In the event of earthquakes, the gas and electricity networks are severely affected. This increases the risk of imminent explosion. To prevent that, any gas leak will be signaled using the built-in gas sensor. The robot is efficient at night, being equipped with a flashlight that can be used both for orientation and for sending signals through Morse code. First aid kit will be attached to it, with medical supplies, drinking water and food.**

**Keywords—rescue robot, natural disasters, android application, remote control**

## I. Introduction

Soon, technology will play a key role in rescuing victims of natural disasters. Search and rescue robots are the main help in this direction. In order not to endanger the life of a victim, either due to injuries or lack of water and food, she needs medical care within 48 hours from the start of the natural disaster. However, it is difficult and dangerous for people to venture into such missions. Rescuers are subject to numerous risks such as explosions, avalanches, drowning, burns or the risk of being trapped under rubble.

To protect rescue teams, engineers came with idea of developing intelligent robots capable of performing, autonomously or remotely guided, these special operations. These robots use high-performance sensors and artificial intelligence algorithms to detect imminent danger in advance and to recognize people who need help [1]. Many research and development institutions have already implemented several prototypes of rescue robots [2].

Japan has an undisputed reputation when it comes to the development of earthquake-resistant buildings. However, the 2011 earthquake in Fukushima had a devastating toll: more than 20,000 dead and more than 100,000 injured. Japan is in a strong seismic zone, which has led Japanese engineers and leaders to find ways to reduce earthquake losses. Good example here is RoboCue, a rescue robot used to recover victims from affected areas. The main tasks of the robot are: collecting information with the help of sensors, creating a safe access road for rescuers and delivering drinkable water and food to people isolated or unable to move [3].

Figure 1. RoboCue recue robot [3].

## II. SYSTEM ARCHITECTURE

This chapter describes the behavior of the two-component system. The hardware component is represented by the rescue robot together with the integrated circuits. The software component is the mobile application that communicates bidirectional with the robot. Block diagram shown in Figure 2 illustrates the connections between hardware modules.

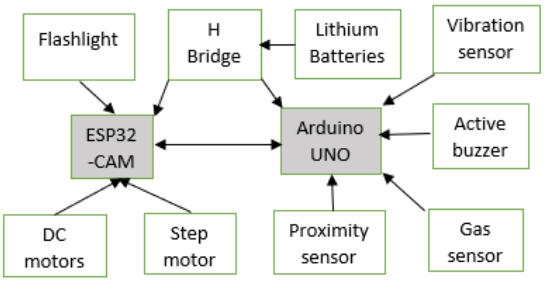



Figure 2. Connections between hardware components.

### A. Process description

Two main interactions are present in this process: robot-external environment interaction and rescuer-robot interaction. Those interactions use two actors: the rescuer and the victim.

Rescuer is permanently connected to the robot via wireless network. It guides the robot through dangerous areas with the help of a video camera. It can receive warning messages from the robot if a gas leak or seismic activity is detected.

Victim is the person who needs help and needs to be found. It communicates indirectly with the robot by signaling its presence.

Robot represents the dynamic part of the system. It has the role of searching for and rescuing victims. It uses the Wi-Fi module to receive commands from the rescuer and is able to detect obstacles or gas leaks. The built-in flashlight is useful for night missions and for transmitting SOS signals. The relationship with the outside is made with the help of sensors and video camera.

### B. *Scenario*

The rescue robot is designed to deal with difficult situations and adapt to all types of natural disasters. In order to provide a simplified picture of the process and a practical understanding of how it works, next paragraph describes a potential scenario that such system could encounter.

A devastating earthquake was recently announced in Fukushima, Japan. Authorities went on alert, firefighters and police officers are mobilized to help the victims. The fire department is equipped with a rescue robot that can be controlled remotely. A team of firefighters is sent to the nearby furniture factory. One of the firefighters approaches the entrance of the building and sends the rescue robot inside. He controls the robot and monitors the received images from inside the building. The enclosure is covered with thick smoke, which makes it impossible for people to move. After a few meters on the room, robot sees a group of isolated employees, who desperately need oxygen masks. The robot detects a gas leak and immediately notifies the outside team to stop the gas supply. Knowing the exact position of the victim group, firefighters can find another access route, accessible and less dangerous for escape.

## III. ROBOT DESIGN

The hardware component of the system is represented by the rescue robot. It consists of several modules, each providing different functionalities. The main features of the robot are remote-controlled movement, live video streaming, obstacle avoidance, gas leaks detection, Wi-Fi communication, night vision and warning sounds. These tasks are shared between two microcontrollers: ESP32-CAM and Arduino Uno.

Arduino Uno facilitates serial communication with a computer or other microcontrollers. With a high quality / price ratio, this microcontroller is compatible with most existing sensors and modules [4]. The ESP32-CAM development board is a small, low-cost microcontroller that incorporates a wireless module. Ideal for IoT (Internet of Things) applications, concept presentations or DIY projects.

The chassis is structured on two levels (Figure 3). On the first level are four DC motors with gear to reduce speed and increase torque. Wheels are made of plastic with a diameter of 10 cm, ideal for robot size.

The four motors can be controlled independently with the L293D motor driver, which facilitates tight turns. On the upper level, the robot has a support for 18650 LI-ION batteries, equipped with a power on / off button.

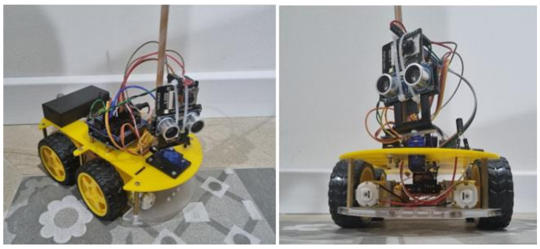

Figure 3. Rescue robot design.

At the front is the camera module, ultrasonic obstacle sensor, gas detector and stepper motor used to rotate the camera. The Arduino Uno development board is placed in the center of the upper floor and has a shield attached to distribute the 5v and GND terminals to the other components.

### A. *Serial communication*

A system is a group of interconnected devices. In order to be able to exchange information, these devices must use the same communication protocol. There are many communication protocols, but they are all divided into two broad categories: serial or parallel. These two types of communication can be compared to two roads, in which serial communication is a national road, with one lane in each direction, and parallel communication is a highway with two or more lanes in each direction. Even if a highway supports a larger volume of cars, the national road can perform the same tasks, but at a cost several times lower [5].

A serial interface sends a single bit of data in a clock cycle. In other words, only two pins are required for asynchronous communication: one for data transmission and one for data reception. This type of communication is very common among interfaces such as USB, Ethernet, SPI or I2C. For synchronous communication, an additional wire is required to synchronize the clock generators. The communication protocol is configurable, the most important thing for the two devices is to share the same protocol.

### B. *Uploading the code on ESP32-CAM*

Serial communication and Over The Air updates were used to load the code. ESP32-CAM does not have a source code upload interface. For this reason, the port of the Arduino Uno board was used. This is only possible by resetting the Arduino board. Another disadvantage of ESP-32 is that it does not have a bootloader (application that facilitates the loading of source code) and requires an additional step: setting the boot-mode status. This state is reached by connecting the ground to the GPIO0 pin. After the flash (source code upload process), this wire must be disconnected.

The whole process is time consuming and can lead to damage to the wires, weakening the contacts. To get rid of this inconvenience during the development of the application, OTA Updates feature had to be implemented. OTA Updates allows you to upload source code to an ESP module using Wi-Fi connection. After adding OTA routines using serial communication, future code updates can be sent using Arduino IDE, a browser, or an HTTP server.

### C. *Movement*

DC motors are controlled using H-Bridge circuit. This module generates PWM signals, which can be further used to adjust the motor speed. PWM (Pulse Width Modulation), as shown in [6], is a rectangular signal used to control the supply voltage of an electronic device in a controlled manner. By switching from logic ‘1’ to logic ‘0’ and vice versa, the signal generates an average voltage equal to the desired voltage. The signal period is closely related to the frequency of the microcontroller's internal clock. For Arduino UNO, the frequency is 500 Hz, which means that the PWM signal period is 2 ms:

$$f = 500\,Hz, f = \frac{1}{T} \rightarrow$$

$$\mathrm{f} = \frac{1}{500\mathrm{Hz}} = 2 * 10^{-3}\mathrm{s} = 2\mathrm{ms} \quad (1)$$

Duty cycle is the percentage of the supply voltage that will be transmitted to the output pin. It is calculated as the ratio between the time interval as the signal is HIGH (5V) and the period calculated according to the relation (1). In this way analog circuits can be controlled using digital signals. With the help of a PWM signal you can control the rotation speed of the motors or change the intensity of a light bulb. A 0% duty cycle is the equivalent of 0V while a 100% duty cycle outputs the unmodified input signal.

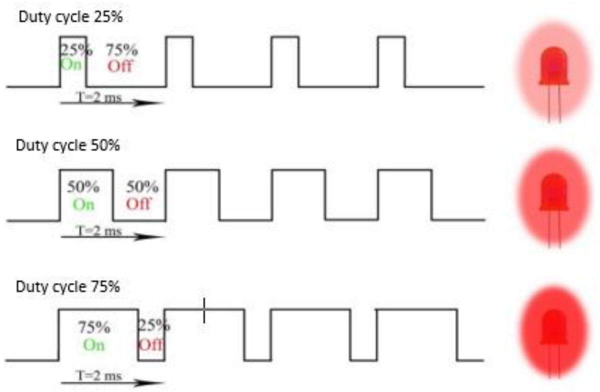


Figure 4. LED response to different PWM signals.

ESP32 microcontroller can control robot moves through the four control pins of the motor driver. The four pins are ENA, ENB, MA and MB. The letter A corresponds to the left side motors, and the letter B corresponds to the right side motors. The ENA and ENB pins are mapped to the GPIO13 and GPIO15 pins and are used to set the rotation direction of the motors. The MA and MB pins are mapped to the GPIO12 and GPIO14 pins of the ESP32 board and are used to set the motor’s speed.

Table 1 shows how the motors work. In order to make a left turn, the right motor will go forward and the left one will go backwards. Similarly, on a right turn, the right motor will go backwards and the left one will go forward.

Table 1. Motors control

| | Speed pins | Speed control | Direction pins | Direction control | |
|---|---|---|---|---|---|
| Motor A | GPIO12 | PWM 0%-100% | GPIO13 | HIGH | LOW |
| Motor B | GPIO14 | PWM 0%-100% | GPIO15 | HIGH | LOW |

## IV. Mobile Application Design

Android is an open source operating system for mobile devices, such as smartphones, tablets or smart watches. It was developed using the Linux operating system by the Open Handset Alliance, a company currently owned by Google.

Android applications are usually developed in Java language using Android Studio development environment. Access to developed applications can be easily done through virtual stores such as Google Play, Opera Mobile Store, or Amazon Appstore.

*RescueBot* is the Android application used to control and communicate with rescue robot. It is developed in the Android Studio environment using Java and XML languages. It is addressed to authorized persons trained to participate in special rescue missions. The appearance of the application is simplistic, without complex design elements that could slow down the execution time of tasks. The focus is on code robustness, handling exceptions, and optimizing tasks. Figure 5 shows the main page of the application, run using the emulator provided by Android Studio. The rescue robot has the camera pointed at the computer monitor.


Figure 5. RescueBot application.

### A. *Implemetation*

Mobile application uses a single activity, MainActivity, which is the "engine" for thread execution. The XML language is used to create buttons, text boxes and for the container that receives streaming data from ESP32-CAM.

The ImageView object occupies the top of the screen and is used for live streaming. When the “CONNECT” button is pressed, the *VideoStream* function is called, which opens an HTTP connection using the *HttpURLConnection* class. The class implements useful functions, the most common being *setRequestMethod* (to set the request type, GET or POST) and *setConnectTimeout* (to set the timeout for connection). Figure 6 illustrates an example of use for these functions.

```
String stream_url = "http://" + ip_text.getText() + ":81/stream";
```

```
HttpURLConnection huc = (HttpURLConnection) url.openConnection();
huc.setRequestMethod("GET");
huc.setConnectTimeout(1000 * 5);
huc.setReadTimeout(1000 * 5);
huc.setDoInput(true);
huc.connect();
```

Figure 6. How to open an HTTP connection.

The created instance uses http://192.168.0.109:81/stream as the connection address. This address contains the unique IP address generated by the ESP32 module (192.168.0.109), the port on which the connection is made (81), and a control parameter that indicates what data is going to be received.

The content of the HTML page is read line by line and converted from a byte vector to a Bitmap object using the Bytes2ImageFile function. This object is then passed to the ImageView object. Figure 7 illustrates how data from the ESP32 camera is received by mobile application.

```
while ((data = br.readLine()) != null)
{
    if (data.contains("Content-Type:"))
    {
        data = br.readLine();
        len = Integer.parseInt(data.split( regex: ":")[1].trim());
        bis = new BufferedInputStream(in);
        buffer = new byte[len];
        int t = 0;
        while (t < len)
        {
            t += bis.read(buffer, t,  len: len - t);
        }
        Bytes2ImageFile(buffer,
                fileName: getExternalFilesDir(Environment.DIRECTORY_PICTURES) +
        final Bitmap bitmap = BitmapFactory.decodeFile(
                pathName: getExternalFilesDir(Environment.DIRECTORY_PICTURES)
        runOnUiThread(new Runnable()
        {
            @Override
            public void run() { monitor.setImageBitmap(bitmap); }
        });
    }
}
```

Figure 7. Processing data received from the ESP32 module.

Figure 5 shows six buttons. Four of them are used to control the trajectory of the robot ("LEFT", "RIGHT", "UP", "DOWN"). The “FLASH” button is used to operate the built-in flash. It can be used to send light signals in Morse code, or for night orientation. Pressing a button calls a function similar to the VideoStream function, but specific to the pressed button.

A SeekBar object is used to move the camera vertically. The object is configured in a range of 1 to 10, one step at a time. Initially the value is 5, which means that the position of the camera is frontal. To move the camera 20° to the right, the search bar must be in position 7. To move the camera 30° to the left, the search bar must be in position 2. The maximum rotation angle of the camera is 100°. The limitation of this angle is due to the hardware connections, more precisely the connecting wires. With the help of *ProgressChanged* function implemented by *OnSeekBarChangeListener* class, a new connection is opened whenever the value of the search bar is changed.

Last design elements are TextView objects. To ensure the connection between the mobile application and the robot, the unique IP address of the ESP32 module must be entered in the editable text box above the search bar. Warning messages sent by the robot are displayed in the text box at the bottom of the screen. Messages such as “Initialize live transmission”, “No danger detected”, “Gas leak detected”, “Seismic activity detected” are displayed in this item.

## V. Development Directions

The solution presented in this research paper is a good starting point for the concept of rescue robot. It can be used in training or simulations, but the construction, hardware limitations and quality of the equipment prevent its use in real rescue missions. Two development directions are described in this chapter, one for each main feature of the robot.

### A. *Resistance and dynamics*

The rescue robot must be able to enter hard-to-reach areas or affected roads. Getting around on wheels is fast, but it is often inefficient, as in the case of fires. The track mechanism is a more suitable solution because it helps to overcome small obstacles and is not flammable. A robotic arm also has a great advantage over the rescue robot because it can be used to clear the path.

The robot body must be weather and mechanical shock resistant. Hardware must be well encapsulated to avoid the risk of short circuit or circuit damage due to high temperatures. Chassis must be light, to improve autonomy, and durable, if areas with a high seismic risk are targeted.

### B. *Searching for victims*

In addition to the video camera and SOS signals, there are many other features that can be added to the robot to improve the search for victims. These include geolocation, a way to determine the location of a smart device. Whether it's a smartphone, tablet, smart watch or laptop, most people have a smart device. Geolocation can be server-based, in which case the IP address to which the device is connected is detected, or device-based, in which case the GPS signal and mobile networks are used [7]. The second method provides better accuracy, especially in overcrowded areas, as it uses data from nearby devices to find the exact location (triangulation). Figure 7 illustrates how the power of mobile networks are used to detect the location of a device. The greater the number of connected devices is, the more it improves location accuracy.

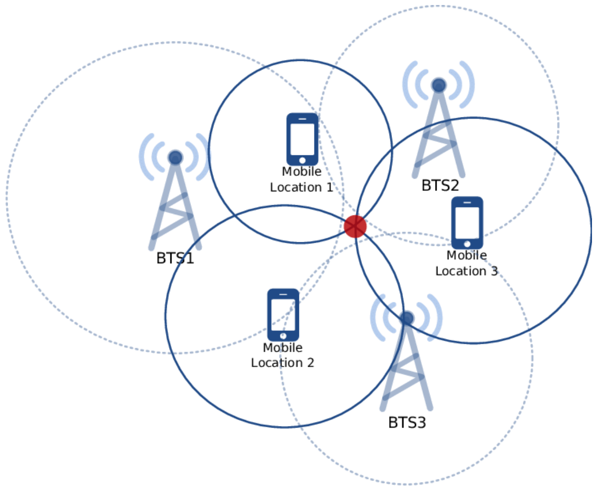


Figure 8. Mobile network triangulation [7].

Equipping the robot with a thermal imaging camera for night missions will be also a great improvement. In this way, people can be detected from far away or under rubble. It can detect objects with high temperatures, with high risk of explosion.

## VI. CONCLUSION

Delays in rescuing victims, inability to find victims or the low number of rescue teams are the main reasons leading to the death of people during natural disasters. In this context, the solution proposed by the disaster control institutions is to develop an intelligent robot to replace human rescuers.

This research paper presents design and development of a rescue robot. The result is an inspirational robot that can be used for further research in this area.

The robotic search and rescue system is a viable concept. It is in its infancy and can receive many improvements. It efficiently uses the processing power of the two development boards, providing a good response time for control commands. It is equipped with sensors for detecting hazards: obstacles, flammable gas leaks or seismic activities. The Wi-Fi connection allows real-time data transmission to both the mobile application and the server.

The mobile application monitors and controls the robot remotely. It is used by authorized personnel, trained to act in extreme situations. The direction of development is to optimize processing speed by reducing non-functional requirements. It works with minimal resources and can be installed on any Android device, including military.

Areas affected by natural disasters pose a major risk to rescuers and rescue time is limited. Given these features, a multifunctional rescue robot based on Wi-Fi and Android technology is a relevant solution. The system integrates the capability of detection, movement and communication and creates an auxiliary rescue platform.

## REFERENCES


[1] Yugang liu, Goldie Nejat, "Robotic urban search and rescue: A survey from the control perspective", Journal of Intelligent & Robotics Systems, Vol. 72, Issue 2, pp. 147-165, 2013

[2] Z. Lin, Z. Yongguo, L. Chengye, "Design of Motion Control System of Rescue Robot Based on ARM", International Conference on Computer Technology, Electronics and Communication, 2017.

[3] DevicePlus Editorial Team, "Robots to the Rescue: Discover the New Generation of Life-saving Robots", deviceplus.com, July 31, 2019.

[4] Microcontrolere, "Arduino Uno", Applications and projects for beginners, Aplicații și proiecte pentru începători, 2016.

[5] X. Jimblom, "Serial Communication",, 2018, unpublished.

[6] J. Heath, "PWM: Pulse Width Modulation: What is it and how does it work?", April, 2017, unpublished.

[7] D. Abodunrin, Y. Miche, S. Holtmanns, "Some Dangers from 2G Networks Legacy Support and a Possible Mitigation", 1st Workshop on Security and Privacy in Cybermatics, 2015.

Facultatea de Automatică și Calculatoare

Universitatea Politehnica Timișoara

**DECLARAŢIE**
**Privind lucrarea ştiinţifică**

Subsemnatul, Robert-Cătălin DIACONU, student în cadrul programului de Master SIAPS, declar că sunt autorul unei lucrări ştiinţifice corelată cu tema lucrării de disertaţie.

Detalii privind lucrarea ştiinţifică (dacă este cazul):
Titlu: Rescue Robot For Natural Disaster Zones

Autor(i): Robert-Cătălin Diaconu

Lucrarea ştiinţifică poate fi încadrată în categoria (bifaţi opţiunea corectă):

☐ publicaţie în revistă
Denumire Denumire revistă, vol. , nr. an , ISSN , alte informaţii , link , pagini ;
Ataşez: ☐ copie după prima pagină, ☐ după cuprins, ☐ copie după articol

☐ publicaţie în conferinţă de specialitate cu program
Denumire Denumire conferinţă, loc de desfăşurare , perioada , link , ISSN/ISBN , pagini , situaţia curentă selectează.
Ataşez: ☐ copie după prima pagină, ☐ după cuprins, ☐ copie după articol, ☐ acceptul pentru lucrare (dacă e cazul)

☒ publicaţie la sesiune de comunicări studenţeşti
Denumire Sesiunea de Comunicări Ştiinţifice Studenţeşti (SCSS-AIA-2022), ediţia 2022 (19-20 mai) - Departamentul Automatică şi Informatică Aplicată, loc de desfăşurare B014, perioada 20 mai, organizator Radu-Emil PRECUP, link
Ataşez: ☒ copie după prima pagină, ☐ copie după programul sesiunii

☐ lucrare validată de coordonator
Nume coordonator L
Ataşez: ☐ copie după articol

Timişoara

Data 11-06-2022

Robert-Cătălin DIACONU