PIBOT
A SERVER CONTROLLED BOT

Dheeraj Kallakuri[1], Vinayak Kurup[2], Madhura Dumbre[3], Sharon laurance[4]

Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai

[1]kallakuridheeraj@gmail.com

[2]kvgkurup201@gmail.com

[3]dumbre.madhura@gmail.com

[4]sharonlaurence@gmail.com

Abstract- This paper discusses the design and implementation of the bot is at a place where a disaster has occurred and the controller of the bot is at a safe place the bot must be controlled remotely. The idea is to provide assistance in an after-disaster scenario. The main objective of this project was to develop a rescue-surveillance framework using minimal technology.

***Keywords*–raspberry pi, Arduino, server, sensors, minimal technology.**

I. INTRODUCTION

Fire hazards, Gas explosion, Building collapse and such urban disasters often lead to widespread damage and those injured can often be very hard to reach for search and rescue(SAR) teams. Robots play a vital role in getting life-saving aid to those in need. There are many SAR bots which are Bluetooth controlled, some implementing transmitter-receiver models, Wi-Fi modules, etc all having a limited range in terms of communication. Also, some bots used nowadays, are remote-controlled whose cost increases with an increase in range. Therefore, the existing technology is quite expensive and thus not every disaster management team can afford it. Our main aim for developing Pibot was to remotely access the bot through servers [2]. Pibot gives the real-time data of its surrounding as well as the live streaming by the camera mounted on the bot. Thus, this implementation of minimal technology makes it less expensive, giving it an edge over existing technologies.

II. BACKGROUND

The controller can operate the bot from a safe location via server and gauge the damage. Real-Time Monitoring is implemented using the web page. There can be two situations. Pibot is better than remote-controlled bot as the latter has a limited range of communication. Our framework has an unlimited range of communication which gives the controller the freedom to operate from a safer place in the event of a calamity. It is able to store sensor data on which data mining algorithms can be used. It needs an Internet connection to process and displays the output. Existing SAR robots[18] like the Tokyo fire department's rescue-bot is designed to locate and safely retrieve victims from the disaster sites, snake bot that is capable of bending and send images back that let the rescuers know what the situation inside the disaster zone is like and many such bots have been implemented. The anomaly of being controlled by remote control and high expenses for such bots can be reduced by implementing the very same framework that we have used for Pi-Bot. i.e. IOT based bot.

III. IMPLEMENTATION

A. Design

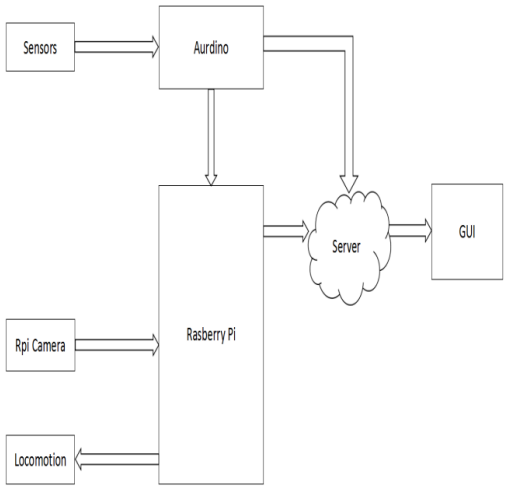


Fig. 2: Block Diagram of Pibot.

As shown in the figure the rpi receives the data from different interfaces and uploads this data on the server. The parent server contains the buttons to drive the bot. on the other hand two-child servers for live video streaming and displaying sensor data. This comprises the GUI display

B. Algorithm

The flow chart below gives the working algorithm for the sensor data.

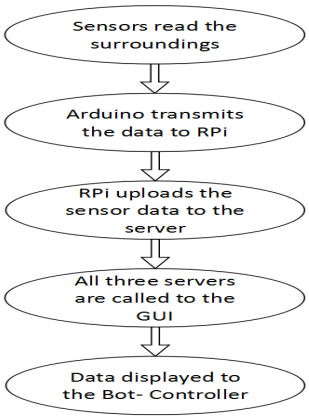
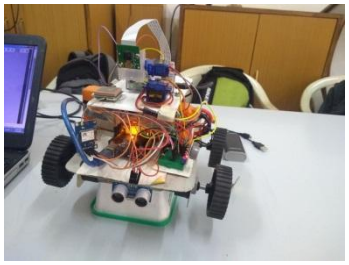


Fig. 2: Flowchart of Pibot

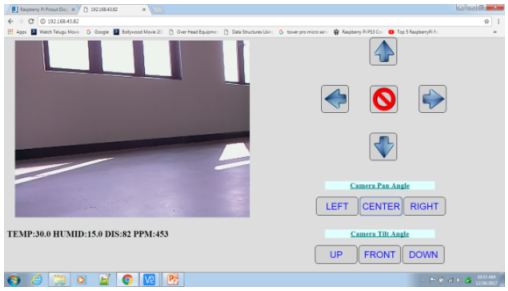
C. Working

PiBot is a prototype for SAR purposes that can operate in inhuman conditions and is capable of performing detailed analysis of the destruction site using its various sensors. The bot is manoeuvred to the disaster-struck site by the controller. The bot becomes the eyes and ears for the control room and gives them vital sensor readings and live camera feed. It can even look for any survivors. The site needs to be shored up and made safe for rescuers to enter which takes up three to four critical early hours of the crisis which are crucial for finding victims alive. The robot has a Raspberry Pi which has a camera kept on a pan-tilt mechanism. There are sensors on the bot which can measure the level of harmful gases and temperature and also tells the obstacles range in its path.



The bot can be operated in 2 modes[10]
1. The controller and bot are on the same network. This is done by using the IP address of rpi that is called on the web browser of the electronic device.
2. The controller is controlling the bot which is in a remote network. This is done using Team Viewer, DNS server.

The RPi hosts a Web Server that has an interactive GUI. The GUI contains buttons [12], on pressing these buttons the python files[14] are called which controls the locomotion i.e. motors and pan-tilt mechanism of the bot. It can also be controlled with the keys of the keyboard. The motors are controlled with the help of motor drivers. WiringPi is a GPIO access library written in C for the Raspberry Pi. It includes a command-line utility "**gpio**" that can be used to program and set up the GPIO pins. Lighttpd [11] is a secure, fast, compliant, and very flexible web server that has been optimized for high-performance environments.



The server for taking live data from sensors has been programmed using NODE.js and JavaScript [15]. GUI has been designed using HTML [3]. Sensors are programmed in Embedded C.
DHT-11 SoftwareSerial.h,tinyGps etc. libraries have been used.

Raspberry pi:

The Raspberry Pi [1][4][5][16] Compute Module contains a BCM2835 processor. The Raspberry Pi is a series of small single-board computers. The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support. The Raspberry Pi 3, with a quad-core ARM processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and the instruction set to use.
This higher-spec variant increases the Raspberry pi GPIO pin count from 26 to 40 pins. There are now four USB 2.0 ports compared to two on the Model B. The SD card slot has been replaced with a more modern push-push type micro SD slot. It consumes slightly less power, provides better audio quality and

has a cleaner form factor. Raspberry comes with only digital pins.

To get started you need a **Raspberry Pi 3 Model B**, a 5V USB power supply of at least 2 amps with a micro USB cable, any standard USB keyboard and mouse, an HDMI cable and monitor/TV for display, and a micro SD card with the operating system pre-installed. The NOOBS (New Out Of the Box Software) OS is recommended Raspberry Pi being mini computer servers can be installed on it. In short Raspberry Pi can do everything that a normal desktop does.

IV. INTERFACING

A. Camera:

The Raspberry Pi Camera [13] Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens.

· 8-megapixel native resolution sensor-capable of 3280 x 2464 pixel static images

· Supports 1080p30, 720p60 and 640x480p90 video

· The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system. The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable.

Applications –

- Motion detection
- Time-lapse photography

B. PAN-TILT Mechanism

The pan-tilt [17] can rotate 180 side to side and 150 upwards and downwards. It is used to mount can cause 2 degrees of freedom motion.

C. Socket programming

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server. Socket programming is done in Node.js. The server is programmed using JavaScript. It is used to take sensor data serially from Arduino. The Arduino is connected to the Raspberry-pi's COM port. Now, this serially transmitted data from the Arduino to the rpi port is directly uploaded to the server.

D. Sensors

1. Ultrasonic sensor

The Ultrasonic Sensor [8] sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone). The ultrasonic sensor gives digital input and output. This is used to find the distance of the obstacle in the path of the bot.

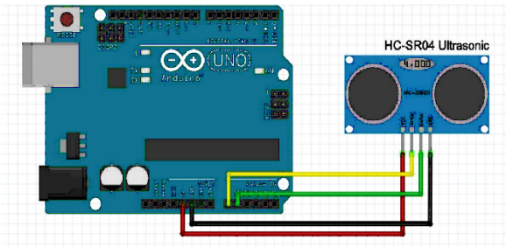


Fig. 3: Interfacing of Ultrasonic sensor with Arduino Uno

2. DHT11

DHT11 [6] temperature range is from 0 to 50 degrees Celsius with +-2 degrees accuracy.

DHT11 humidity range is from 20 to 80% with 5% accuracy. They consist of a humidity sensing component, an NTC temperature sensor (or thermistor) and an IC on the backside of the sensor. It uses digital pins.

The surrounding temperature and humidity of the bot are dynamically measured and displayed on the web page.

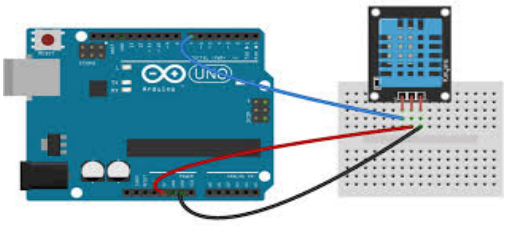


Fig. 3: Interfacing of DHT-11 sensor with Arduino UNO

3. MQ-7

The MQ-7[7] can detect CO-gas concentrations anywhere from 20 to 2000ppm. This sensor has high sensitivity and a fast response time. The sensor's output is an analogue resistance.
CO is a very harmful gas to humans. The MQ7 Gas Sensor module is useful for gas leakage detection.

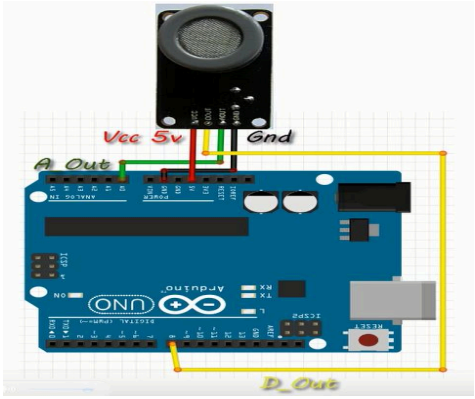


Fig. 3: Interfacing of MQ7 gas sensor with Arduino UNO

4. GPS Sensor[9]

GPS sensors are receivers with antennas that use a satellite-based navigation system with a network of 24 satellites in orbit around the earth to provide position, velocity, and timing information. It uses digital pins. This is useful to find the exact location of calamity.

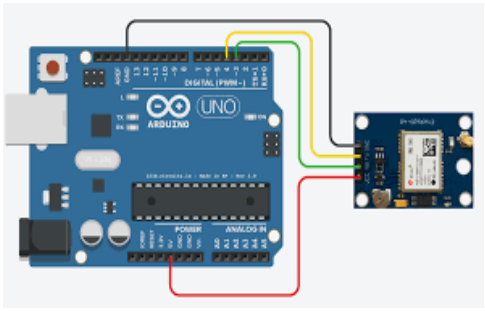


Fig. 3: Interfacing of GPS sensor with Arduino UNO

V. CONCLUSION

The Pibot assists in search and rescue operations by remote access and control and can be deployed by various esteemed services. PiBot is a step towards better security and efficient rescue operations by the use of minimal and low-cost technology. Pibot is a prototype that can be further developed by performing appropriate changes. Futures add on will include adding automated rescuing capabilities but the basic framework of using servers will remain the same.

VI. ACKNOWLEDGEMENT

We would like to acknowledge Mrs Smita Dange, Coordinator of Computer Department, FCRIT, Vashi for her guidance on this project as well as providing us with the required hardware. We would also like to thank our family for being a constant pillar of support.

REFERENCES

[1]https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html
[2]https://www.dexterindustries.com/BrickPi/brickpi-tutorial-documentation/projects/browserbot/
[3]https://www.w3schools.com/html/
[4] https://static.raspberrypi.org/files/legacy/qsg.pdf
[5]https://www.raspberrypi.org/products/raspberry-pi-3-model-b
[6] http://theelectronics.co.in/dht11-interfacing-arduino/
[7] https://www.youtube.com/watch?v=ji7JbwmdlJo
[8]https://www.theengineeringprojects.com/2015/02/interfacing-ultrasonic-sensor-arduino.html
[9]http://arduinostuff.blogspot.in/2014/05/neo6mv2-gps-module-with-arduino-uno-how.html
[10]https://pihw.wordpress.com/guides/direct-network-connection/
[11]http://www.instructables.com/id/IoT-Controlling-a-Raspberry-Pi-Robot-Over-Internet/
[12]https://www.w3schools.com/jsref/event_key_keycode.asp
[13]http://blog.ramon.no/2013/06/build-simple-internet-controlled-webcam.html?m=1
[14]https://app.assembla.com/wiki/show/oniontv-plugin/Using_python_in_the_web_with_lighttpd_server
[15] https://www.youtube.com/watch?v=DNeHg4eIB6w
[16] https://www.raspberrypi.org/
[17]http://www.instructables.com/id/Pan-Tilt-Multi-Servo-Control/
[18]https://www.popsci.com/technology/article/2011-03/six-robots-could-shape-future-earthquake-search-and-rescue