

Multipurpose Espionage Robot for Military/Police Use

Vishal Khemnar*, Ishan Pandita**, Sarita More***, Rahul Bhandari****, Prof. Amol Dumbare*****

*(Computer Engineering, PCCOE&R, Ravet, Pune
Email: vtckhemnar@gmail.com)

** (Computer Engineering, PCCOE&R, Ravet, Pune
Email: ipandita6@gmail.com)

*** (Computer Engineering, PCCOE&R, Ravet, Pune
Email: saritamore27@gmail.com)

**** (Computer Engineering, PCCOE&R, Ravet, Pune
Email: iamrahulbhandari2015@gmail.com)

***** (Computer Engineering, PCCOE&R, Ravet, Pune
Email: amol.dumbare@pccoer.in)

Abstract:

In many situations, for spying and surveillance purposes, human lives are put in danger. It may be an army officer who has to retrieve sensitive information by breaking into an enemy base. This involves penetrating foreign territory which can lead to the capture of that officer which is dangerous for his life. Another danger is that some techniques could be used to extract information from the captured officer instead. This can prove dangerous to the concerned nation. Other such scenario could be in bomb detection. An army officer will have to risk his life to locate a bomb hidden at an unknown location in a huge area. Landmines are also responsible for taking many human lives. In these scenarios, if we had robots that could do the dangerous work in place of humans, it could really be helpful.

The robot proposed by us is built in two modules- Lower Movement Module and Upper Control Module. The upper module will be used to control the camera and to stream the live video captured by the camera onto a Wi-Fi network where other connected computers can see the live video. The lower module is a movement module which will move the upper module from one place to another as instructed by the user. The advantage of this is that the movement module can be changed according to the terrain that we want the robot to move in. For example – hilly areas, plains, highways, water and ice will all require different movement modules whereas the upper module will remain the same.

Keywords —Camera, Espionage, Security, Surveillance, Raspberry-Pi, Wi-Fi, Arduino, Robot, PC.

I. INTRODUCTION

Robots have been used from many years by humans to perform their jobs with efficiency and in lesser time. It is obvious why we are so obsessed with robots as they make our life easy and enjoyable. Some argue that robots are rendering humans useless and soon will take over the world,

but keeping those fictional stories aside they do aid us in many activities which are physically and mentally impossible for human bodies like manufacturing thousands of cars in a single day. Robots can assemble, paint and test a car with complete autonomy. Another example can be drones that give us a top view of our surroundings

within a minute. They can be used for surveying and spying on the surrounding areas.

The system proposed by us is a robot that can be used for surveillance in areas where human life could be in danger or in areas where it isn't possible for humans to enter. It can go into small spaces like tubes, bore wells, etc. It can go in areas with hazardous gases in the air and give updates through an attached camera. The robot moves with the help of wheels which can steer the robot right, left, back and forward. The wheels have a belt over them to help the bot maintain traction and move in uneven areas.

II. TECHNOLOGY USED

The following technology is used in making the robot:

- 1) Arduino UNO
- 2) 200RPM 12V Motors
- 3) L293D Motor Driver
- 4) HC-05 Bluetooth Module
- 5) 12V Lead-acid Battery
- 6) Raspberry-Pi 3B+
- 7) Pi-Camera
- 8) TP-Link TL-WR740N 150mbps Wi-Fi Router
- 9) Python language (to program the Raspberry-Pi)
- 10) C language (to program the Arduino UNO)
- 11) VLC Media Player

The Arduino UNO takes commands via the HC-05 Bluetooth module on how to move the robot. The Motors are controlled with L293D motor driver chip which also helps give power from the 12V lead-acid battery to the motors. The Arduino is programmed in C language using the Arduino IDE. This makes up the lower movement module.

The Raspberry-Pi has a Pi-Camera attached to it which captures the video in front of the robot. Raspberry Pi also has on-board Wi-Fi which we use to connect to the Wi-Fi router to make it a part of the network. The Raspberry-Pi can be programmed in Python without the need of a separate IDE.

The streamed video can be watched on any computer with VLC Media Player installed in it and which is connected to the same network as the Raspberry-Pi.

III. WORKING

The Raspberry-Pi executes a command in Python on start-up as follows:

```
raspivid -o - -t 0 -hf -w 800 -h 400 -fps 24 |cvlc -  
vvv stream:///dev/stdin --sout  
'#standard{access=http,mux=ts,dst=:8160}' :demu  
x=h264
```

- -o Specifies the output filename. the '-' beside denotes no filename
- -t is the duration of the recording, 0 being infinity
- -hf is Horizontal Flip
- -w and -h is the resolution for Width and Height
- -fps is Frames per Second
- The rest means that on port 8160, data will be sent through http using h264 as stdout using VLC

This will start recording the video and stream it over the Wi-Fi network. Now we connect a PC with VLC Media Player installed to the network and navigate to: VLC -> Media -> Open Network Stream -> Network -> Please enter a network URL. It will open a screen like this:

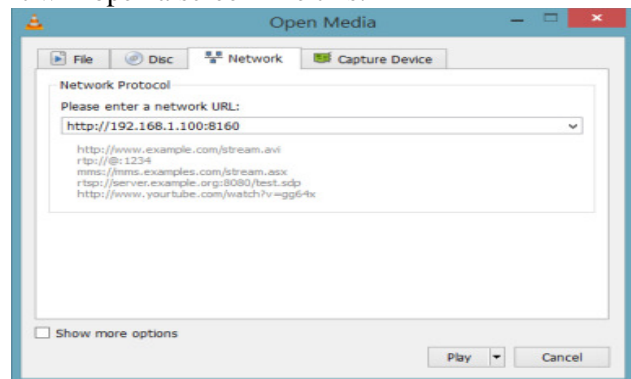


Fig.1. VLC Media Player

Here, we enter the IP address of the Raspberry-Pi on the network followed by ‘.’ and then by the ‘PortNumber’ which in our case will be 8160. To keep the IP address same each time, we reserve the IP address for the Raspberry-Pi in the router’s settings. After clicking on the ‘Play’ button, the live video from the Pi-Camera will show up on the computer as long as both the devices are in range of the router.

IV. RESULTS

The risks that were identified are mentioned below:

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Battery Damage	High	Low	High	Low
2	Internal Short-circuit	Low	High	High	High
2	Chassis Bending	High	Low	High	Low

Fig.2. Identified Risks

Probability	Value	Description
High	Probability of occurrence is	> 80%
Medium	Probability of occurrence is	261 – 79%
Low	Probability of occurrence is	< 20%

Fig.3. Risk Category Percentages

Risk ID	1
Risk Description	Description 1
Category	Power Failure.
Source	Identified during testing.
Probability	Low
Impact	Low
Response	Accept
Strategy	Backup Battery Option
Risk Status	Occurred

Fig.4. Risk 1

Risk ID	2
Risk Description	Description 2
Category	Hardware
Source	Previous similar projects’ study.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Better insulation during component installation.
Risk Status	Identified

Fig.5. Risk 2

Risk ID	3
Risk Description	Description 3
Category	Robot Body
Source	This was identified during early testing phase.
Probability	High
Impact	Low
Response	Mitigate
Strategy	Use stronger material for chassis.
Risk Status	Identified and Occurred

Fig.6. Risk 3

The video frame-rate on the receiver was very low (15fps) but works. The resolution of the video was 800x400. The 12V 2 Ah battery powered the four motors for around 30 minutes. None of the identified risks occurred during the testing period.

V. CONCLUSIONS

The robot for surveillance purposes presented works as intended with a video frame-rate of 15fps, 800x400 resolution and a battery backup of around 30 minutes in one charge.

ACKNOWLEDGMENT

We are very much obliged to our project guide Prof. Amol Dumbare and Head of Computer Department, Prof. Dr. ArchanaChaugule in Computer Engineering Department, PCCOE&R, Ravet, Pune for helping us and giving us proper guidance.

REFERENCES

- [1] [Online].Available: <https://tutos-sphinx-test.readthedocs.io/en/latest/source/rpi/video/video-streaming-vlc.html>
- [2] [Online].Available:

- <https://create.arduino.cc/projecthub/electropeak/the-beginner-s-guide-to-control-motors-by-arduino-and-1293d-139307> [4] Banzi, Massimo; Shiloh, Michael. Getting started with Arduino (Third ed.). Sebastopol, CA. ISBN 9781449363314. OCLC 898290173.
- [3] "Arduino Software Release Notes". Arduino Project. Retrieved September 25, 2019 [5] "Python 3.8.0". Peterson, Benjamin (19 October 2019). "Python 2.7.17 released". Python Insider.