International Journal of Computer Applications (0975 – 8887)
Volume 180 – No.1, December 2017

Terrestrial Robotic Security Surveillance Rover to Enhance Security at the International Borders and Buildings

Nyambutora Malaki Simon
School of Computing and Informatics
University of Nairobi, Kenya

William O. Okelo-Odongo
School of Computing and Informatics
University of Nairobi, Kenya

Muchemi L. Githiari
School of Computing and Informatics
University of Nairobi, Kenya

ABSTRACT
Terrorists are finding new ways of attacking developing countries like Kenya using current technology. Security forces and guards are exposed to high risk of attack due to lack of modern facilities to monitor international borders and buildings. Due to this challenge, a system was proposed that would be used to monitor these places remotely. This system was able to interact with other security gadgets like drones to effectively survey international borders and buildings, send real-time alerts, and stream videos live from anywhere. An Application Programming Interface (API) was developed to interconnect the system with a drone. A web page and a simple Android application were developed to stream videos and control the rover over the internet. Raspberry Pi 3 Model B and the Arduino UNO microcontrollers were interconnected and used to run the Rover. Satellite, Wi-Fi, Internet and the GPRS technologies were used for communication. Appropriate sensors were mounted on the rover to collect information from the environment and relay real-time data to the central server for further analysis and interpretation. The system was tested, evaluated, and recommendations made for future work.

General Terms
Security Surveillance

Keywords
Terrestrial, Robotic, Microcontroller, Rover, Architecture.

1. INTRODUCTION
Security authorities are getting involved in using information technology to do surveillance, communicate and respond to crimes [1]. In the study by Robert and Daniel [1], information technology safeguards the country’s borders, including the management of how individuals and goods get into and out of the country. However, this is not the case with developing countries like Kenya. In these countries, the military and the security personnel are directly risking their lives in efforts to counter terrorism. Such countries have embarked on ways to carry out surveillance most of which are not successful yet.

It is unfortunate that when a country is attacked, the authorities have little to do to help the terror victims. For instance, the current security situation in Kenya is worrying. Even though Kenya has been at the forefront in the war against terrorism for decades, this does not mean they have had experience in dealing with terrorists. This fight against terrorism doesn’t seem to be so easy for Kenya. The study shows that in years Kenya has experienced countless threats, if not attacks dating back in the mid-1970s [2].

A serious terror attack, back in 1998, at the American embassy in Nairobi, left many Kenyans and the international community shocked after it claimed the lives of 200 people, 12 of whom were Americans. [2] The incident left thousands injured as well raising dire need of Kenya to seriously embark on counterterrorism strategies to save its people. Despite these efforts to counter terrorism, the country again was hit by a serious terrorist attack in 2002 when a hotel was bombed by an al-Qaida-affiliated group [2]. These and many more cases have been reported. An example is the recent killings at the Westgate Shopping mall in Nairobi in 2013 [2] that claimed 67 lives, and the murder of Garissa University students in which 147 students lost their lives to terrorists [3].

It is obvious that when such kinds of attacks happen, the Kenya security forces are the ones in direct danger of facing the terrorists. The Kenyan government has lost so many soldiers in such scenarios and even leaving so many injured. For instance, fighting terrorists inside buildings is not an easy task; failure to manage it properly using relevant technologies may lead to the loss of security personnel and civilians.

Therefore, proper and appropriate techniques must be in place. These techniques are achievable using appropriate technology that can help survey buildings and even the international borders. For instance, the border between Kenya and Somalia must be safeguarded from intruders like the Al Shabab. People are aware that Somalia government is unstable because of the Al-Shabab and their internal conflicts. Therefore, Kenya and other countries must protect its international borders to avoid such attacks and loss of lives. This protection can also be achieved using appropriate technology that is cheap and one that the Kenyan government can afford considering its poverty level [2].

2. RELATED WORK

2.1 Autonomous ground vehicle for distributed surveillance
According to Ashitey and John [4], an autonomous ground vehicle for distributed surveillance can replace human beings in highly security sensitive areas. In their work, they suggest on the integration of various components in their future work for their system to be able to perform some tasks simultaneously. However, developing such a system may not be easy considering the complexities involved in the developments. This requires proper designs and innovation techniques.

2.2 Robotic vehicle control using the internet via webpage and keyboard

In their study [5], this system would have an increased area of surveillance due to the use of the internet. It had an arm that would pick and place objects as appropriate. However, the proposed robotic vehicle used only the raspberry pi, which was limited in the number of the GPIO pins. Since processing of videos by the raspberry pi requires a lot of memory, another microcontroller would have been used to control the robotic arm to reduce overhead on the Raspberry Pi.

2.3 Android controlled robot with image transfer

The robot was able to take images from its surrounding, and transfer to a connected system with the aid of Bluetooth technology. Since the camera was available and mounted on the vehicle, this made it better from the traditionally fixed cameras for surveillance [6]. According to Ritika and Narender [7], the use of Bluetooth has changed the way people use the devices at their home places and at offices. Bluetooth has further helped in the conversion from wired to wireless devices. The only challenge with using Bluetooth is that it is limited within a radius. This calls for an option to use the internet to avoid radius limitations.

2.4 CCTV cameras

Governments have installed CCTV cameras to try and monitor borders and buildings. However, CCTV cameras can only be placed in a fixed position within the areas of interest. Some areas get undermined because of the fixed cameras. [8] Argues that when there are more cameras within a premise, they tend to push wrongdoings like a crime to other areas. He continues to say that the surveillance cameras on their own cannot keep off crime and that they may only assist on some other crimes within their reach, making other crimes shift to other places or get displaced.

3. METHODOLOGY

This project proposed a four-wheeled rover capable of monitoring at different terrains, and at the buildings. The study involved the development of a prototype that informs in the development of robust surveillance systems. In the subsections that follow, the project proposes the user-rover interaction architecture, the high-level design architecture, the robot design architecture, the drone-rover integration architectures, the flow of events within the rover, and the different interconnected components.

3.1 User-Rover Interaction Architecture

The figure below shows how users (security personnel), interact with the terrestrial rover.

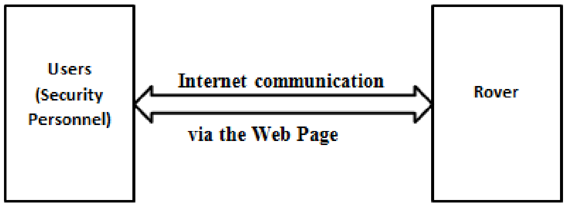

Fig.1 User-Rover Interaction Architecture

Communication between the rover and the user was via a web page with the help of the internet. It is on this web page where commands that instructed the rover on what to do was triggered.

3.2 High-level Design Architecture

The diagram in the figure below shows the various interconnections between the devices. The computer and the smartphone (Android phone) were both used to control the Rover from within the Local Area Network and also from the internet.

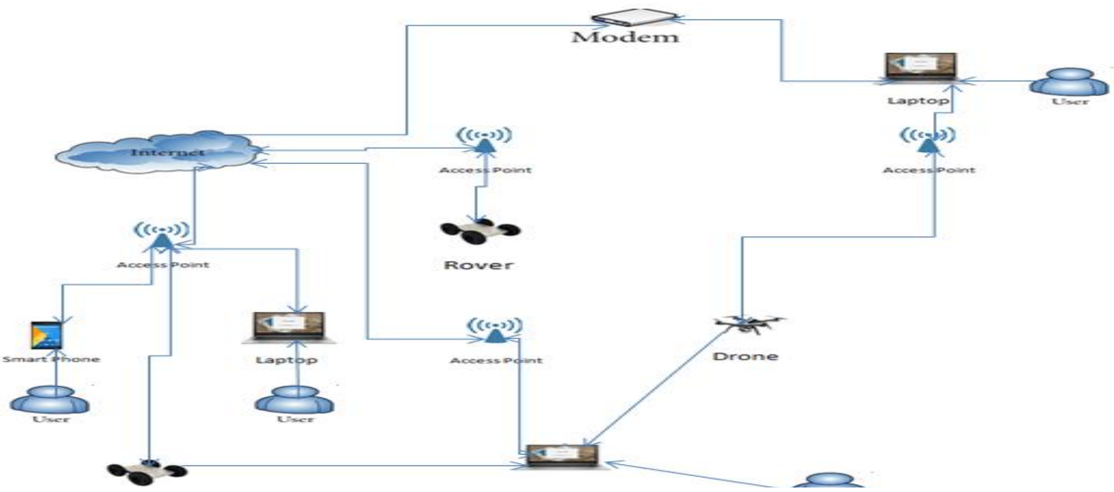

Fig.2 High-level Design and the interconnections by author

Using the Dataplicity application on the Android phone, someone would ssh the Rover to do basic configurations. A simple Robot Rover application was developed was used to control the Rover via the internet. Once the Rover was powered from the LiPo battery, the Arduino code was loaded into its Arduino UNO Microcontroller via the Raspberry Pi which was Wi-Fi-enabled. Videos captured by the Rover's Raspberry Pi camera were relayed directly to the web page on either the computer or the Android phone. To send GPS information to the central monitoring system, the GSM/GPRS/GPS module was used. This module was connected to the Arduino UNO, and with the help of an Arduino C programming language, a sketch was written to be able to commit GPS information from the Rover direct to the central monitoring system.

The figure above also shows how a third-party gadget, the drone (externally controlled and automated), was connected to the central server. Using the same technique used with the Rover, the GSM/GPRS/GPS module was attached to the Arduino UNO connected to the Drone. The collected information was sent to the Central Monitoring System. In both cases, sending of data to the central system was made possible by the web service developed to do so. Otherwise, communication between the drone and the Rover would not have been possible.

If one wanted to switch off the Rover, he or she would simply **ssh** the Rover and write the command **sudo halt** on the *lxterminal* of the Raspberry Pi or **sudo reboot**, if he or she wanted to restart the Rover.

3.3 Robot Design Architecture

The architecture of this robotic system was as shown in the figure below:

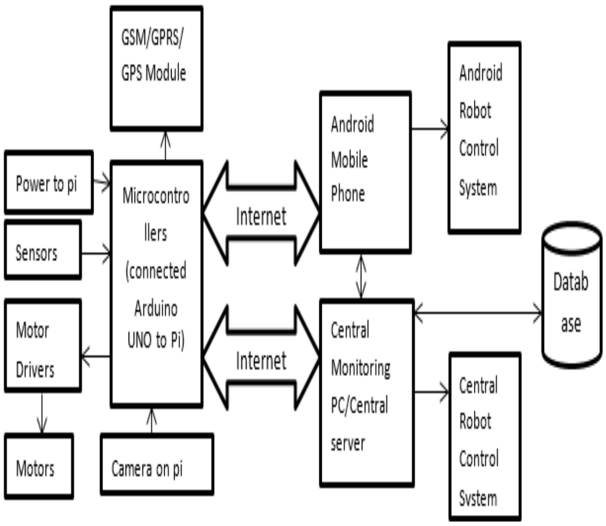

Fig.3 Robotic Design-Architecture by the author

In this project, control of the rover was over the Android phone and the central server personal computer using the internet. The robot collects information from the environment and saves into the central server or a local PC. It captures and sends live video streams using the internet to the Android phone and to the primary PC for appropriate actions to be taken. For instance, security-text messages can be triggered by the click of a button to designated individuals for action to be taken.

3.4 Integration Design Architecture

The designed robot system interconnected with another surveillance gadget (drone) as the architecture shown below:

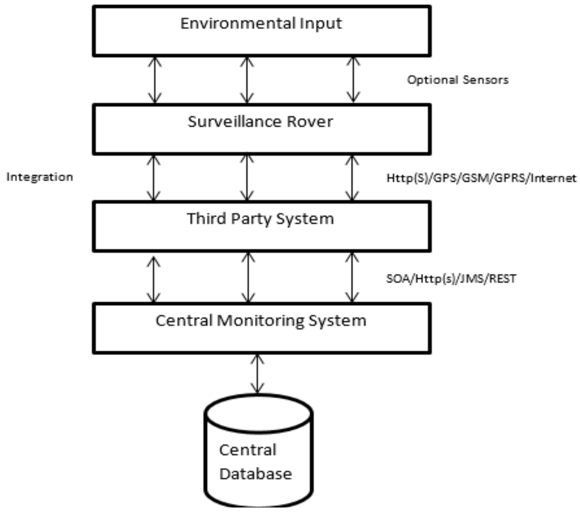

Figure 4: Integration Architecture by author

Using appropriate technologies such as HTTP(s) and SOA, the Rover was interconnected with other external gadgets like a drone, which in this case was a third-party system. This interconnection was made possible using a central monitoring system and a central database.

3.5 Flow of Events

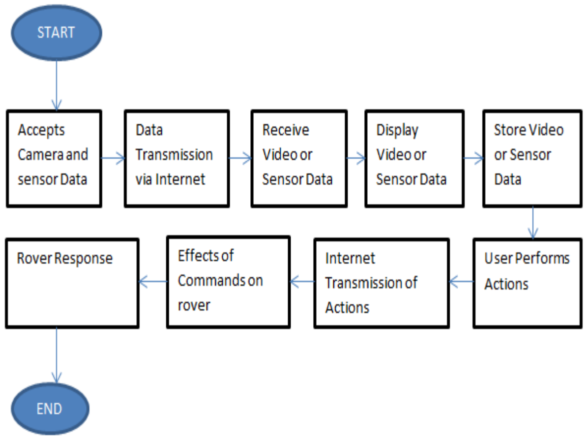

Fig.5 Flow Diagram by author

The flow diagram above shows the flow of events from the time of starting the rover to the point the rover stopped. In the first stage, the streaming videos were captured by the Pi Camera. These videos then were transmitted via the internet to the central computer on an Android device where they are received in the form of a display or saved on the hard drives.
By the kind of information received, real-time information, the user performed actions which were transmitted via the internet and which finally triggered some effects on the rover. The user was able to see the rover respond accordingly. The cycle was repeated and again until when the Rover was turned off.

3.6 Connection of Components

As shown in the figure below, communication between Arduino UNO and the Raspberry Pi 3 was by a USB cable. Data from the Arduino UNO was sent to the Raspberry Pi serially in bits of 0s and 1s. These bits contain the required information. Raspberry Pi 3 communicates by sending characters commanding the Arduino UNO on what to do. Consequently, the Raspberry Pi was the 'master' handling complex tasks and the Arduino UNO was the 'slave' handling simple tasks. This method of connecting the two devices was the best because it provided unlimited options. The devices were able to share the load and more pins were available to attach the sensors.

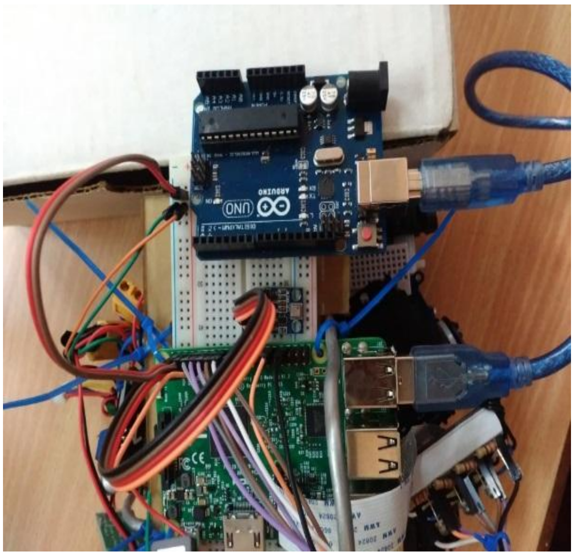

Fig.6 Arduino UNO and Raspberry Pi 3 Connections

The GSM/GPRS/GPS module shown below was attached to the Arduino UNO. The module was used to send rover coordinates to a server installed with the drone-rover API to be able to identify the locations of both devices.

Fig.7 gsm/gprs/gps

The Pi-Camera was used in this project to stream live videos from the Rover. Then, the videos viewed on a web page. It was mounted on the Raspberry Pi and extended to the front of the pan and tilt to facilitate movement in various directions. The figure below shows how the LEDs were connected next to the Pi-Camera to be able to provide light in the case of darkness.

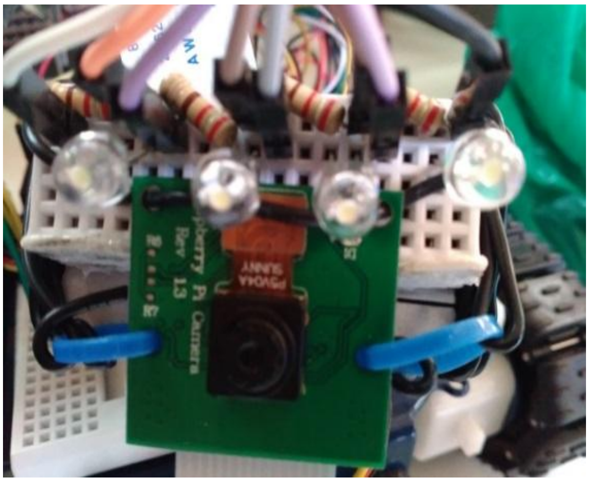

Fig.8 Pi-camera

The **SN754410NE Motor Driver** from the Texas Instruments was used in this project because it controls the DC motors in all directions. In simple terms, it is a bidirectional motor driver. It has the similar pin out just like the L293D commonly used in the robotic world. Nevertheless, it has the extra benefits of protection diodes that are inbuilt. It requires a voltage range of 4.5V to 36V to operate.

The project also uses the Solarbotics gear motors which are inexpensive. The gear motors go at 66RPM using the only 5V. The power provided by the motors was enough to drive the rover around. In addition, it was easier to control the speeds of the gear motors.

3.7 Software and Platform

3.7.1 Arduino UNO and Software

Arduino UNO has a development platform that is easy to use and easy to understand. It has a text editor on which to write the code and some other functional features. Some of them are the console, the buttons in the toolbar section containing menus on each and the messaging area among other features. This environment provides communication and allows the uploading of written codes to the Arduino UNO hardware. Note that this environment is installed in the Raspberry Pi because the Raspberry Pi is just like a computer. However, one can as well develop using an external computer like a laptop and then load the program to the Arduino UNO.

In this project, serial communication was used to communicate between the Raspberry Pi and the Arduino UNO. The function used to be Serial.available(). For instance, characters ‘a,' ‘b,' etcetera were used as commands. Whenever such characters are triggered either from a web page, some instructions get executed like moving the rover forward, backward, etcetera. No instruction is executed in a loop unless there is a trigger either from the input signal on the Arduino UNO. The project also used the PhpSerial library to be able to send commands from a web page to the Arduino UNO, which later controls the movement of the gear motors and the other features attached to the Arduino UNO. The sketch below shows how a program was written to allow the robot to be commanded from the web page.

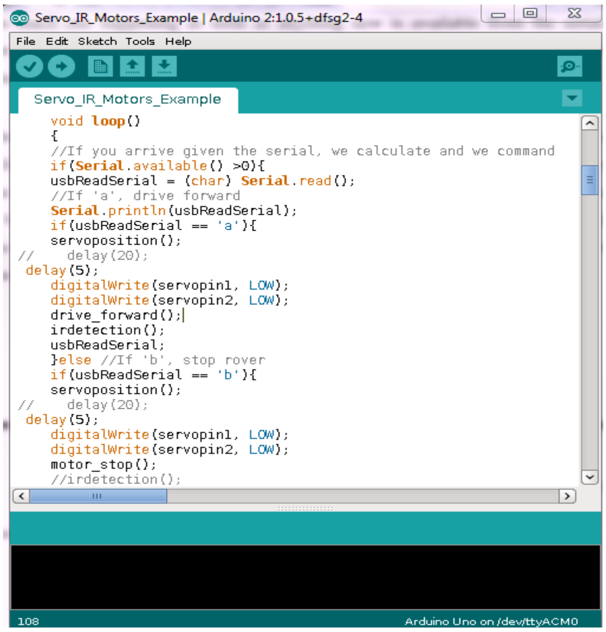

Fig.9 Serial Communication using characters as commands

It is from the above platform that one puts his or her code to run the GSM/GPRS/GPS shield. This shield as mentioned earlier, it is to allow communication between the rover and the drone. The project used a 4G GSM Sim card that acted like a modem. Since the code for running the module was continuously in a loop, data from the shield was sent to the central server at constant intervals. This data helped locate the device at a real time. Note that, the code had to load into the hardware. In this project, the sketch program is used to run the gear motors, the pan and tilt, the IR sensor, and the GSM/GPRS/GPS module.

3.7.2 Raspberry Pi and Software

If one configured the Raspberry Pi to be accessed wirelessly, then the manually configured IP address was removed to avoid problems during boot. The command **sudo nano /boot/cmdline.txt** was used to access the cmdline.txt file and remove the IP address.

The raspberry Pi development environment was suitable for the development of the project. Someone had to install the latest version of Raspbian Jessie version 8 Operating System.

Open source software was installed on the Raspberry Pi for different functionalities. First, one had to install LIGHTTPD web server. LIGHTTPD was a brilliant idea in this project simply because it is a more lightweight web server. Unlike Apache, it is faster and does not involve too many features. It uses fewer resources, in contrast to the Apache.

Second, PHP scripting language was used substantially in this project. Whenever one run a page from the web pages, PHP script got executed. The execution was done remotely on the server, which in this case was the LIGHTTPD.

Third, the project required the use of Python within the Raspberry Pi. This programming language was easy and straightforward to use because it had a simple syntax that entailed only a few lines of code. Python is the most common language used with the Raspberry Pi. In addition, JavaScript was used to make the work look smart, especially the user interface.

Fifth, Wiring Pi was installed to help write a shell-script that helped send commands from a page to control the GPIO pins of the Raspberry Pi.

Lastly, the project used libraries like the mjpg-streamer library (to help run the camera), Adafruit-bmp085 Library and much more as it was required by the various components and manuals purchased with them.

In this study, the use of crontab assisted in the scheduling of activities or recording of data collected from the environment continuously after certain time intervals. By typing **sudo crontab –e** command, one gets a file to edit to ensure various activities are scheduled as needed.

4. RESULTS

4.1 User Interface

A web form was created such that only authorized and registered users could log into the system. This page was a secret page that was only made available to the administrator. Rover users were required to surrender their names, email addresses and password. Once registered, a user would log into the system either locally using the WLAN or from the internet and stream live videos and control the rover. Rover control could be done in different directions and at different speeds.

Buttons were set to control the gear motors to allow the user to control the speed of the rover once clicked. A section for light control and sending alert messages was also useful. In the case of darkness, the user could use the light buttons to light up a place to be able to provide clear videos. If the user detected or observed some weird behavior from the video like detecting an intruder, messages were sent to alert the relevant authorities by the click of a button.

Fig.1 Rover Control, Monitoring and Alert Page

As shown in the figure above, the pan and tilt camera section allowed users to be able to twist the camera at various angles to be able to survey an area with ease. The pan and tilt were set to specific angles, though this could be altered depending on the user's needs. There was also an option to view the temperature readings or even log out of the system.

4.2 Internet Control

The surveillance rover was controlled from any Android device over the internet using REMOT3.IT by Weaved, Inc. Remote.it is much more secure than port forwarding. The software was installed on the Raspberry Pi. It was useful in cases where one wanted to control more than one Pi via the internet. Dataplicity assisted in accessing the Raspberry Pi from anywhere on the web. Since most internet service providers block UDP packets for security reasons, someone had to contact his or her internet service provider for permission to transmit UDP packets.

4.3 Android Control

Using a simple Android application, the rover was controlled by an android phone. Once logged in, the user would perform any duties just like it would be executed from a PC. This research project took advantage of Dataplicity android application to connect to the Raspberry Pi and to SSH from anywhere. The figure below shows an android application interface.

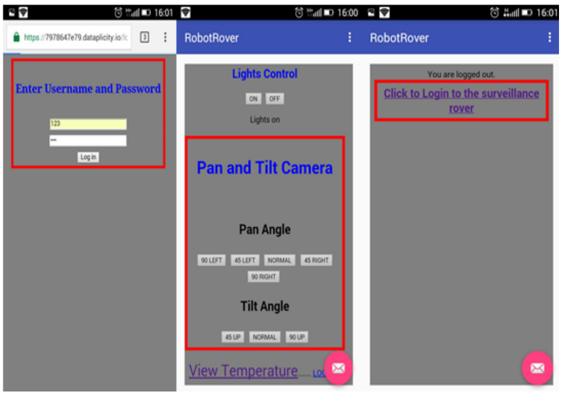

Fig.2 Simple Android Application Interfaces

4.4 Video Streaming

The images 'a' and 'b' in the following figure show the screenshots of live video streams taken by the Rover from within the building (indoor) and outside of the building (a simulated environment) respectively.

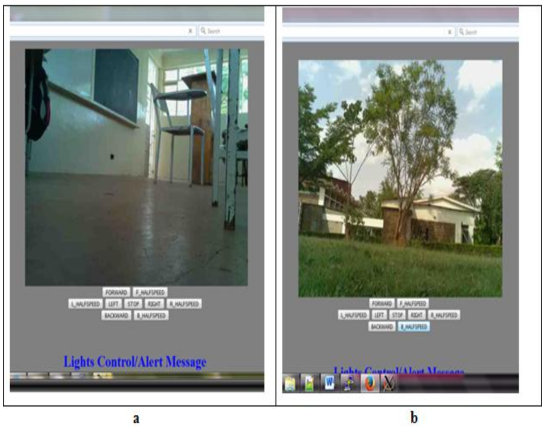

Fig.3 Rover Streams both in the building and in a simulated environment

4.5 Video Download

It was possible to capture and download the video externally with a steaming machine. The video was recorded using the available software applications like the Movavi Screen Capture which one would purchase at an affordable fee. Once the video was captured, it was stored on an external drive for later use. The figure below shows a video captured inside a building using the same software.

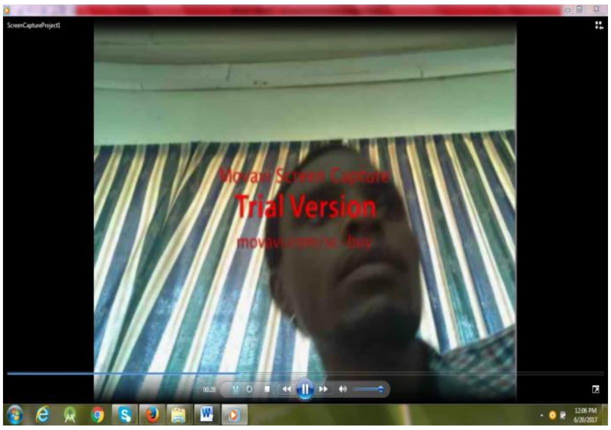

Fig.4 Movavi Screen Capture Video

4.6 Drone-Rover Integration

The Drone-Rover API web service developed using REST interconnected the drone and the rover. With the help of the API, data were obtained from the Arduino UNO's attached GSM/GPRS/GPS module and sent to the central server for storage and future retrieval. Data from the drone and the rover was committed to the same central server. It was the same data that were used by the central monitoring system to give location information about the drone and the rover. The coordinates like the longitude and the latitude assisted in identifying the rover's name. The drone and the rover developed are shown in the figure below.

Fig.5 Drone and the Rover

4.7 Drone-Rover Monitoring Center

As shown in the figure below, the drone-rover monitoring center assisted in retrieving stored data at the central server. It was from this interface that devices would be monitored to be able to keep track of their positions.

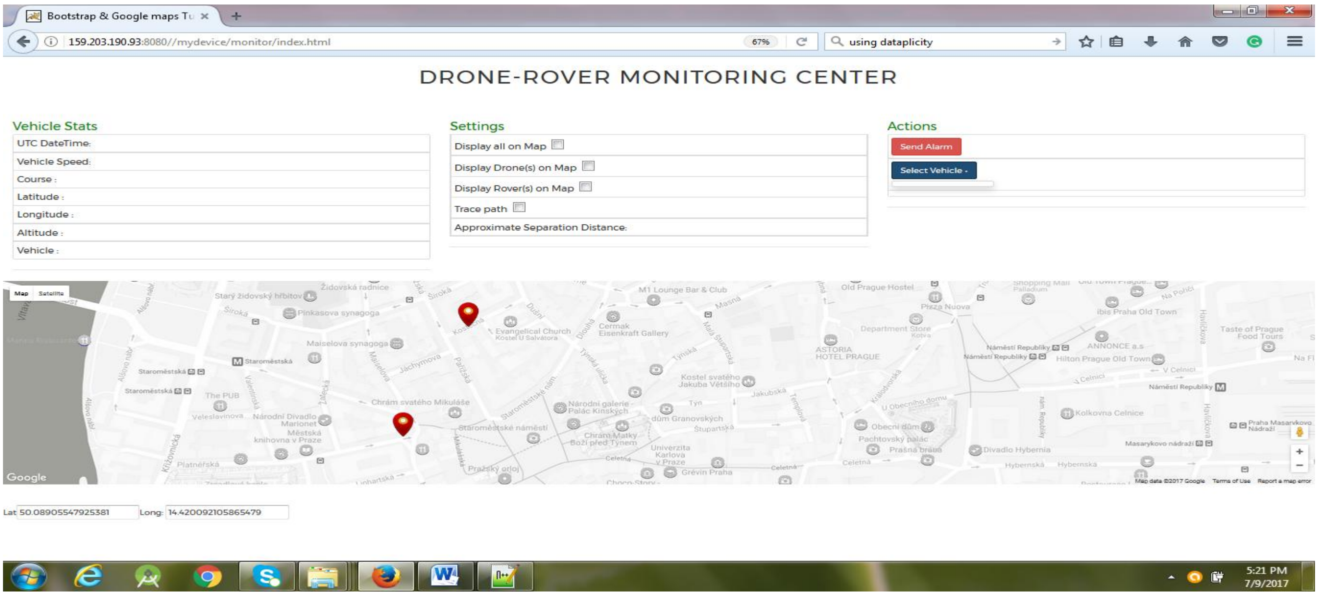

Fig.6 Drone-Rover Monitoring Center

As from the figure above, an authorized individual could select a vehicle of choice from the drop-down box highlighted blue. This drone-rover monitoring center would then enable the authorized personnel to be able to see the vehicle statistics (in this case, the vehicle is either the drone or the Rover) which included the UTC DateTime, vehicle speed, course, latitude, longitude, altitude and the name of the vehicle. Also, various settings were done from the same interface. These included, display all on the map (all available vehicles), display Drone(s) on the map, show Rover(s) on the map, and finally, trace paths taken by vehicles. The Google Map feature showed the respective vehicle location on the map for easy tracking. It is the same interface that an alarm can be triggered to inform security personnel of any problem. The sent email message contained information about the location of the respective vehicles. This interface made the surveillance task simple and manageable from a central location.

5. DISCUSSION

5.1 Data Collection and Automation

The project's main data collection components were the IR sensor, the BMP180 pressure sensor, and the GSM/GPRS/GPS module. The IR and the module were both mounted on the Arduino UNO while the BMP 180 was connected to the Raspberry Pi. The IR sensor was calibrated to detect objects roughly within 10 centimeters. A function was created that reads the distance between the rover IR sensor and the detected object. Whenever an object is detected to be within that range; the Rover stops moving in whichever direction. Once it stops, the camera twists in the specified direction (in this project, it twists to the left) and then the servos refreshed. The rover can also move backward to take a different direction as specified and rotate the camera accordingly. This control of the code makes the rover automated to some degree. In the case of an obstacle, it can easily avoid preventing the destruction. This method of automation makes the rover safe in case the user forgot to control it from the commands available on the web page.

The BMP180 pressure sensor is useful because it detects and informs the users of the changes in environmental temperature and pressure. Data on the current temperature and pressure is collected and committed to the database at regular intervals of one minute. The collected data is then retrieved and displayed on a page for analysis by the user. This information is so important because any abnormalities on the graph generated may be detected by the user and then necessary action taken. The work is made easier for the user because immediately the Rover is powered on, the Pi camera starts streaming videos. Scheduling can be done to allow the Pi camera stream videos for a duration of time. Crontab makes work even easier to achieve the proper scheduling of tasks.

5.2 Architecture Communication and Performance

The design architectures used were the most appropriate. The project used two major architectures that involved the connections of Raspberry Pi and Arduino UNO. The two devices use serial communication technology. Communication from the web page to the Arduino UNO gets simplified by the utilization of the PhpSerial library. Someone had to assign a button to a character to be able to trigger a command to the Arduino UNO. This approach avoids too much complexity on the side of code development.

This study used Raspberry Pi and Arduino UNO to share the load. Raspberry Pi was used to stream videos, collect BMP180 sensor data, provide light to the rover in the case there is some darkness and finally, command the Arduino UNO to do various tasks. This way, work was shared which saves on processing time, and resources hence improved Rover performance.

Integration between the rover and the drone forms another important architecture. It is through such architecture that the rover would communicate with the drone. The project provided a web service capable of collecting relevant information from the two devices. This web service acted as a center for communication between the two devices. This transmission was possible using the GSM/GPRS/GPS module. The Central Monitoring System is used to provide an interface where tracking of both the drone and the rover was accomplished. The authorities controlling the drone would use the coordinates from the rover and perform an automated mission of locating the rover.

5.3 Human Factor Consideration

For the project to achieve its performance levels, the project engaged some individuals who gave their contributions to the main decisions and requirements. Based on the kind of information they provided, right decisions on the kind of hardware and software used were made. This project would have been of no use to clients (the security personnel and even other customers) if all stakeholders were not involved. Students, security personnel, and other relevant individuals were engaged. By analyzing their needs, a system that best reflects current demands on technology to curb security vulnerabilities was implemented. A user-friendly graphical user interface was implemented. This interface performed well as shown in previous figures, but with only a few challenges. For instance, at some point, one had to double click on the buttons for the servos to be triggered. This was because of bad resetting on the side of the Arduino UNO. However, the user would easily interact with the user interface and control the Rover with a lot of ease. Human involvement led to the introduction of new methods and techniques in every part of this project. Novel approaches resulted in the kind of system at hand that can inform the development of a robust system which can be used to enhance security.

5.4 Cost Evaluation

The use of available open source hardware and software makes the project cheap and affordable to the end user. The choice of equipment utilized by the developer is best because they are cheap and some readily available locally. Products used in the development of the prototype came in bulk, which drastically reduces the cost, hence making the end product cheap and affordable to the users. The fact that this is an internet and Android controlled project brings its price far below. Furthermore, Android phones used are readily available and owned by almost everyone.

6. CONCLUSION

Considering the high rates of terrorism and insecurity at the international borders and buildings, countries must embrace current technologies to curb the risks associated with insecurity. International borders and buildings can be monitored by using simple and cheap technologies developed through innovation. Systems such as rovers and drones can be integrated to work together and accomplish one mission of surveying these areas. Remote surveillance can be achieved using rovers and drones, which avoids putting the lives of security personnel and security guards in danger.

7. FUTURE WORK

The study suggests:

- Studies and innovations on similar rovers that are more automated to remotely carry out security surveillance in sensitive areas.
- Studies in the development of object detection Rovers which detect objects and move towards such object to capture videos and images.
- Developments of the same type of rovers and drones, but using different types of microcontrollers for comparison and improvements.

8. ACKNOWLEDGEMENT

I would like to thank God for the gift of life and good health that I was able to complete this project. I also thank the late Prof. William Okelo-Odongo who was always available for supervision and guidance until his demise. I extend my appreciation to Dr. Muchemi for the support and advice. Finally, I thank my classmate and friend Mr. Fred Marube Ondieki whom we tirelessly worked with to ensure that we had a working drone-rover API for integration purposes.

9. REFERENCES

[1]. Robert and Daniel 2008. Digital Quality of Life: Understanding the Personal and Social Benefits of the Information Technology Revolution. [Online] Available from: http://www.innovationfiles.org/digital-quality-of-life/ [Accessed 16 July 2015].

[2]. Samuel Aronson, L. 2013. "Kenya and the Global War on Terror: Neglecting History and Geopolitics in Approaches to Counterterrorism". African Journal of Criminology and Justice Studies: AJCJS, Vol.7, #s1 & 2.

[3]. BBC News 2015. Kenya attack: 147 dead in Garissa University assault. [Online] Available from: http://www.bbc.com/news/world-africa-32169080 [Accessed: 7th July 2015].

[4]. Ashitey and John. 1999. An Autonomous Ground Vehicle for Distributed Surveillance: CyberScout. A technical report by the School of Computer Science at Research Showcase @ CMU. Robotics Institute. Pittsburgh, PA, 15213, USA.

[5]. Dumbre et al. 2015. "Robotic Vehicle Control using the Internet via Webpage and Keyboard". International Journal of Computer Applications (0975-8887). Volume114-No. 17, March 2015.

[6]. Saravana Kumar K. et al. 2015. "Android Controlled Robot with Image Transfer". International Journal of Advanced Multidisciplinary Research 2(3): (2015): 98–102.

[7]. Ritika and Narender 2014. "Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller". International Journal of Scientific Engineering and Resea rch (IJSER). Volume 2 Issue 7, pp 14-17.

[8]. Adrienne, 2001. Can Surveillance Cameras Be Successful in Preventing Crime and Controlling Anti-Social Behaviors? Paper presented at The Character, Impact, and Prevention of Crime in Regional Australia Conference convened by the Australian Institute of Criminology and held in Townsville 2-3 August 2001.