

# Surveillance Robot Using Arduino Microcontroller, Android APIs and the Internet

Durgesh Chandraker<sup>1</sup>, Shirpad Desai<sup>2</sup>

<sup>1</sup>Student, Department of Electrical Engineering, Bharati Vidyapeeth (Deemed to Be University), College of Engineering, Pune, India

<sup>2</sup>Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth (Deemed to Be University), College of Engineering, Pune, India

**Abstract** - We propose a cost-effective four-wheeled surveillance robot using an Arduino UNO microcontroller and a smartphone running the Android Operating System. Surveillance robots typically consist of a video camera, a GPS module, and GSM radios. Android smartphones come with excellent hardware satisfying the above needs. This can be leveraged and used to advantage through APIs (Application Programming Interfaces) provided for the Android operating system. Moreover, the cost for building said robot using a smartphone is mitigated to a great extent. The robot can be controlled remotely from a PC using the internet and a microcontroller-smart phone interface residing on the robot. To capture and archive the real time video from the robot, the inbuilt camera input of the phone is utilized. The robot can be controlled based on visual feedback from the same smart phone. Four motors help achieve a zero turning radius. The camera is attached to a stepper motor which makes it feasible to capture the scene or object of interest. The captured video can be enhanced and made intelligible using further image processing on the remote PC thereby eliminating the need for extra DSP hardware on the robot.

## I. INTRODUCTION

Surveillance is the process of monitoring a situation, an area or a person. This generally occurs in a military scenario where surveillance of borderlines and enemy territory is essential to a country's safety. Human surveillance is achieved by deploying personnel near sensitive areas in order to constantly monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always possible. There are also added risks of losing personnel in the event of getting caught by the enemy. With advances in technology over the years, however, it is possible to remotely monitor areas of importance

by using robots in place of humans. Apart from the obvious advantage of not having to risk any personnel, terrestrial and aerial robots can also pick-up details that are not obvious to humans. By equipping them with high resolution cameras and various sensors, it is possible to obtain information about the specific area remotely. Satellite communication makes it possible to communicate seamlessly with the robots and obtain real-time audiovisual feedback. Thus, in recent times, surveillance technology has become an area of great research interest. However, building a small robot for testing and research purposes proves to be extremely expensive. Primarily because a security robot would require certain components such as a GPS module (Global Positioning System), High resolution cameras, radios for satellite connectivity, etc. Each of these components are quite expensive and piecing them together for the purpose of a robot is a very costly and time-consuming affair. Moreover, a lot of time is wasted in writing driver code to interface all these components. The solution to this dilemma is quite simple. In the last few years, feature-rich smartphones have become popular. These phones come equipped with the required features such as a GPS module, a high-resolution camera and internet connectivity. Due to the extremely efficient supply chains that go into manufacturing consumer electronic devices, these phones come quite cheap for the features that they provide. Also, the operating system on these smartphones provides Application Programmer Interfaces (APIs) for using the various sensors with ease. By using the APIs provided, we can easily write apps in a high-level language like Java, without the complication of writing driver code. In our system, we have used a smartphone running the Android

Operating System [8], which is one the most popular mobile operating systems today.

Thus, it is our aim to build a fully featured surveillance robot using an easily available Android phone, which can be remotely controlled over the internet.

## II. RELATED WORKS

The field of surveillance robots is quite popular. A lot of work has been done in navigational algorithms and control system of wireless surveillance robots [1], [2]. A common theme is also the use of a camera on the robot in order to receive live video feedback [1], [3], [5]. Wireless robots made using the Arduino microcontroller have been implemented, but wireless communication occurs using the Zigbee protocol, which limits the range of the robot [6]. A robot which performs image processing using the camera on an Android smartphone has also been implemented [7]. However, this method is limited by the processing power of the phone, a problem that we have addressed by remotely performing all imaging processing operations on a different computer, after transmitting the camera's feed.

Our project is rather unique in the sense that it is a low-cost solution that offers the ability to remotely control a robot with an unlimited range (by using the internet), while also offering video feedback. There is also no constraint on any extra processing since everything is done remotely.

## III. SYSTEM DESCRIPTION

Our system consists of a remote computer and a robot. The robot is controlled by a user sitting at the remote computer, over the internet. The robot consists of a smartphone running the Android operating system, an Arduino microcontroller to control the robot's motion, and the requisite hardware (motors, chassis, power supply, etc.)

The user controls the robot by sending control signals to the Android smartphone. The smartphone then forwards these signals to the Arduino Microcontroller, which then moves the robot in the required direction. The camera on the Android smartphone is used to send video feedback to the remote user simultaneously over the internet. This enables the user to navigate the robot remotely. Additional processing can be performed on

the video feed on the remote computer. A visual representation is shown below.

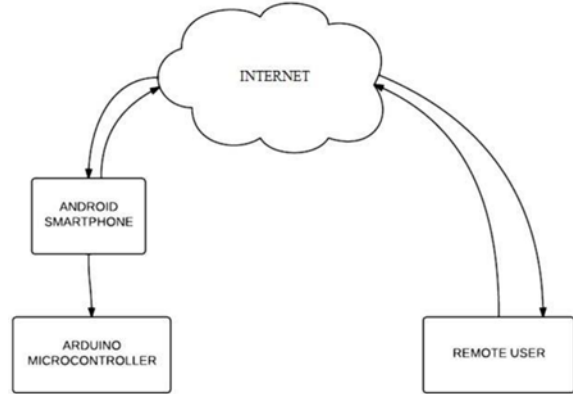


Fig. 1 A basic overview of the system.

## IV. MODULES AND INTERFACES

The project is divided into 3 modules, each of which are explained in detail below.

### A. Android to user Communication

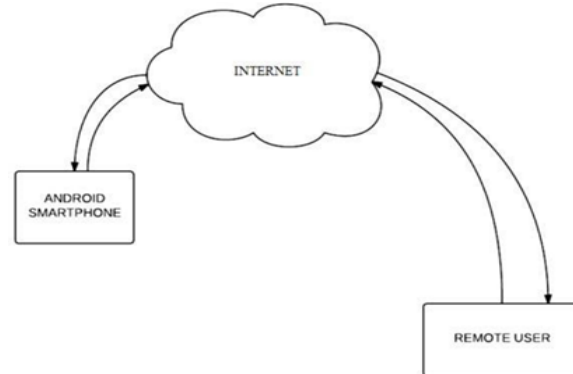


Fig. 2 Android to user interface.

At the remote computer, the user can control the robot through a MATLAB GUI that we created using the GUIDE toolset [19]. A screenshot of the same is shown below.



Figure 3 MATLAB GUI on the remote computer

As seen in the figure, there are 4 navigation buttons to move the robot. There is also a window showing the live video feed from the robot. The slider above the video feed allows us to control the brightness of the incoming video. We have also implemented an auto contrast feature for the video. The ability to record, save and play back the video feed has also been implemented.

The Android smartphone on the robot writes the video feed to a particular IP address which is set by the user. When the user clicks on the “Start Video” button, a MATLAB timer is set off. This timer calls back on a read function at regular intervals of time. This function reads the image that is transmitted by the Android phone at that instant in time. The image is then passed through the user-controlled brightness and contrast filters, and then displayed in the GUI. The interval at which the read function is called defines the frame rate of the video feed. A record feature is also implemented where in the sequence of RGB matrices of the frames are stored in a 4D matrix. This matrix is stored as .mat file, which can be played back later. Navigation of the robot, based on the video feed, is done using the buttons on the GUI, or through the arrow keys on the keyboard. When a particular button/key is pressed, a unique String assigned to it is sent to the same IP address. This String is read by an app on the Android smartphone.

Both the video feed (phone to user) and navigation signals (user to phone) occur simultaneously on different ports of the same IP address. This is done with the help of internet sockets [14].

Figs. 4 and 5 show flowcharts depicting the functioning of the GUI and the video feed respectively.

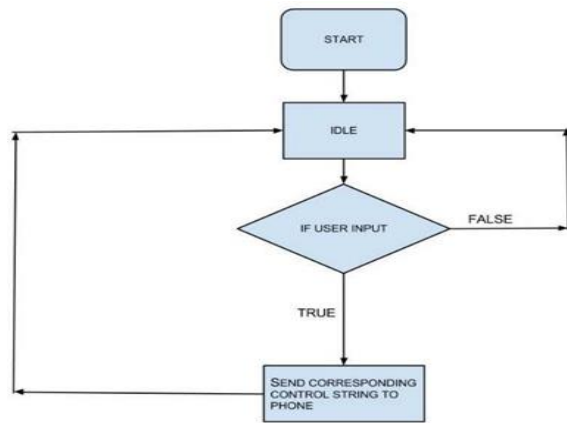


Figure 4 User Input through MATLAB GUI

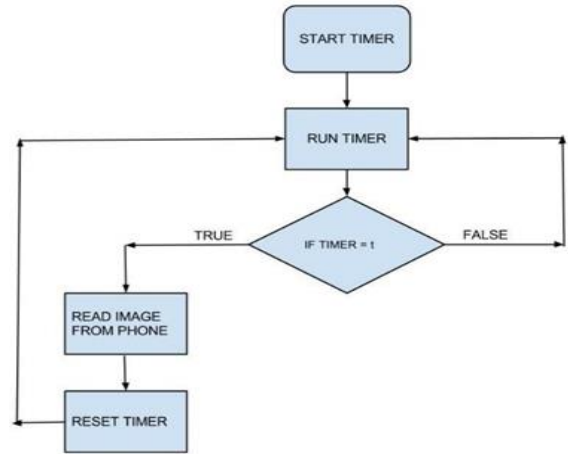


Figure 5 MATLAB function for reading transmitted video.

### B. Android to Robot (Arduino) Communication

The Android phone is connected to the Arduino using a USB OTG (On-the-Go) cable. Android provides support for USB connected devices through two modes of communication: USB host and USB accessory. In our case, the Android phone acts as the host and powers the Arduino [9].

The Android phone acts as a bridge for communicating between the remote computer and the Arduino. For this purpose, there are two separate apps running simultaneously on the phone. One of these is for transmitting the video feed to the remote computer. For this purpose, we have used a freely available Android app called IP Webcam [18]. The second app is responsible for receiving control signals from the Remote User and relaying them to the Arduino microcontroller. A screenshot of the app is shown below.



Figure 6 A screenshot of the Android app running on the phone

The app is coded in Java using the Android Software Development Kit (SDK) [10]. The app listens for incoming Strings from the Remote User on a specified port. Each String is then mapped to a hexadecimal value. For example, an “up” command would be mapped to 0x00, “back” as 0x01 and so on. This hexadecimal value is then sent to the Arduino microcontroller. We have used the USB Host API provided by Android for this purpose [9].

In the app, we have also included five navigation buttons for testing purposes. These buttons simulate the same response as the ones on the MATLAB GUI.

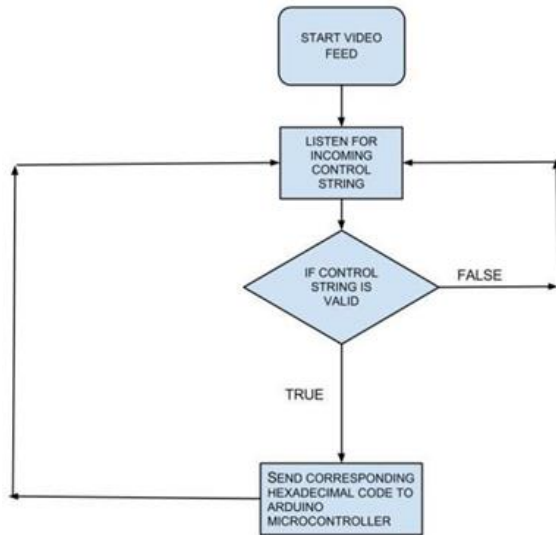


Figure 7 Workflow of the Android app on the smartphone

### C. Arduino Implementation

In our system, we have used an Arduino Uno is a microcontroller board based on the ATmega328 [11]. The Arduino project provides an integrated development environment (IDE) based on Processing, and programming is done using a language based on Wiring, which is very similar to C++ [13].

The Arduino microcontroller is configured to receive serial input from the app running on the Android smartphone, and subsequently control four DC motors (2 front and 2 rear).

Upon receiving the hexadecimal codes from the Android phone, the Arduino generates two control signals per DC motor. For e.g. on receiving 0x00 to indicate a forward motion, the code on the Arduino sends one HIGH and one LOW on each pair of control signals. A backward motion would involve inverting of the same, and so on. Since the Arduino cannot directly power a DC motor due to insufficient current,

motor drivers, with their own power supply are used. Each motor driver is capable of controlling 2 DC motors. Hence, two motor drivers are used. In our implementation, the Arduino sends the control signals to two L293D motor drivers each powered by a 9 volt battery [17].

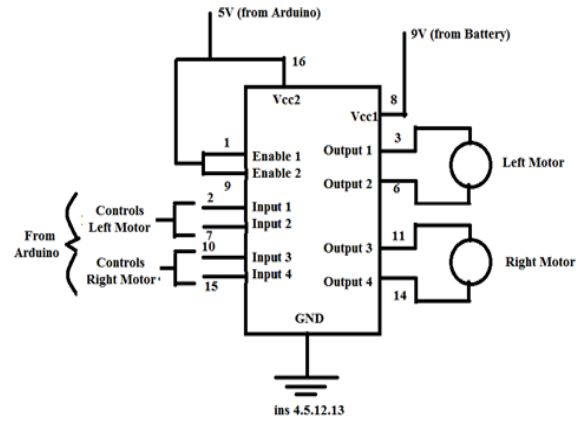


Figure 8 A circuit diagram of the motor driver circuit

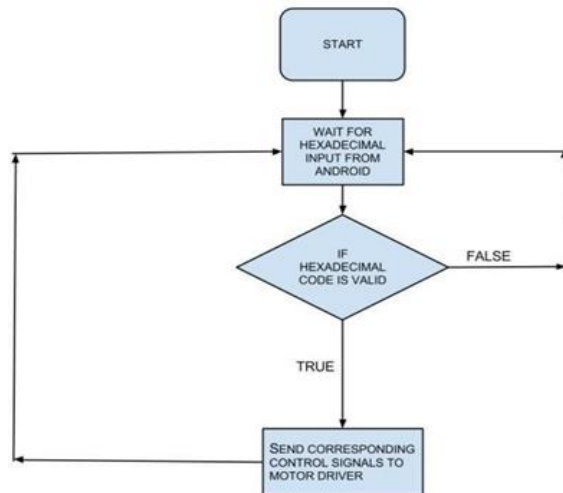


Figure 9 Flowchart depicting functioning of Arduino

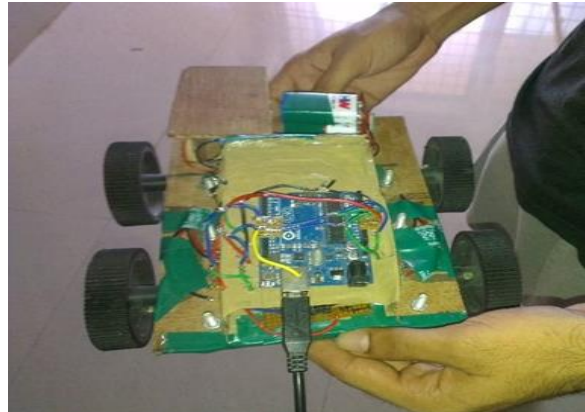


Figure 10 A picture of the robot, with the Arduino on top

## V. SCOPE FOR FUTURE WORK AND CONCLUSION

This project offers a lot of scope for adding newer features. Since all image processing is done remotely, there are no resource constraints apart from the bandwidth of the network. We can program the robot such that it can detect objects and reach them on its own. Thus, we can make it completely autonomous. Also, with the presence of GPS navigation and mapping software, the robot has the capability of finding the best route possible to reach a certain location. Also, by making it sturdier and giving it extra protection, we can make it an all-terrain robot, which would make it ideal for a surveillance robot.

There is also the option of adding sound processing to the remote computer, thus giving it greater surveillance capabilities.

The possibilities are endless. This robot in its current state provides a platform for further research into improving its capabilities.

## REFERENCES

- [1] Hou-Tsan Lee, Wei-Chuan Lin, Ching-Hsiang Huang, Yu-Jhih Huang, "Wireless indoor surveillance robot", in 2011 Proceedings of SICE Annual Conference (SICE), 2011, p. 2164- 2169
- [2] Change Zheng, "Mechanical design and control system of a miniature surveillance robot", in ICIA '09, International Conference on Information and Automation, 2009, p. 1228- 1233
- [3] Kyunghoon Kim, "Intelligent surveillance and security robot systems", in IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), 2010, p. 70- 73
- [4] D. Matko, "Image based control of a space surveillance robot", IEEE International Conference on Robotics and Biomimetics (ROBIO), 2011, p. 2838- 2843
- [5] Ki Sang Hwang, Kyu Jin Park, Do Hyun Kim, Sung-Soo Kim, Sung Ho Park, "Development of a mobile surveillance robot", in ICCAS '07, International Conference on Control, Automation and Systems, 2007, p. 2503- 2508
- [6] Christian Hernández, Raciél Poot, Lizzie Narváez, Erika Llanes and Victor Chi, "Design and Implementation of a System for Wireless Control of a Robot", IJCSI International Journal

- of Computer Science Issues, Vol. 7, Issue 5, September 2010
- [7] Mejdil Safran and Steven Haar, "Arduino and Android Powered Object Tracking Robot". [Online]. Available: [http://faculty.ksu.edu.sa/mejdil/Publications/Android\\_Arduino\\_Robot.pdf](http://faculty.ksu.edu.sa/mejdil/Publications/Android_Arduino_Robot.pdf)
- [8] Android Developers Site [Online]. Available: <http://developer.android.com/index.html>, 2012
- [9] Android Developers Site on USB Host [Online]. Available: <http://developer.android.com/guide/topics/connectivity/usb/host.html>, November 2012
- [10] Android Developers Site API Guides [Online]. Available: <http://developer.android.com/guide/components/index.html>, October 2012
- [11] The Arduino Uno Website [Online]. Available: <http://arduino.cc/en/Main/arduinoBoardUno>, September 2012
- [12] Arduino Language Reference [Online]. Available: <http://arduino.cc/en/Reference/HomePage>, September 2012
- [13] Shiffman, Daniel (September 23, 2009). "Interview with Casey Reas and Ben Fry". Rhizome.org