

# Design of a Remote Control System Based on a Mobile Platform for Indoor Environment

<sup>1,\*</sup>Quan-Ming Zhao, <sup>1</sup>Pei-Pei Jin and <sup>1</sup>Tao Zhang

## Abstract

Nowadays, equipment used for environment monitoring and video surveillance is commonly placed in fixed positions in the indoor environment, which results in the limitation of monitoring scope. Thus, this paper illustrates a design of a remote control system based on a mobile platform for indoor environment. As the mobile platform of the system, the smart car based on STC89C52 MCU is controlled remotely by an Android client through wireless communications. An OpenWrt wireless router and a network camera are attached to the smart car, so the Android client can get video content and temperature of any point where the smart car arrives in real time. According to the video, the motion states of the smart car can be changed by the Android client, such as the velocities and directions. After testing, the system can achieve the function of monitoring temperature and video surveillance with high flexibility and extendibility.

**Keywords:** remote control, smart car, Android, video surveillance, OpenWrt

## 1. Introduction

Smart homes have emerged with the development of electronic information technology, automation technology and computer networks [1], which aim at making home life more comfortable, safe and convenient through integrated management. Varieties of traditional devices can be connected to the Internet, such as smart TV, air conditions or smart sensors.

Computers, tablets, smart phones and robots are widely used as interfaces in smart homes [2-6]. However, smart phones are increasingly promoted as a choice for its convenience to carry compared with the others [7-8]. A new survey of IDC reveals that the shipments of Android smart phones has increased to 14.3 billions in 2015, which has a global market

share of more than 82%. The Android OS [9] based on the Linux OS is open for equipment makers and programmers, so they can customize the phone as much as they like [10]. Therefore, a variety of applications have been written to extend the functionality of Android phones based on the Java programming language. The applications developed for the remote control are widely used in smart homes, such as lighting devices, TVs, hi-fi sound system and computers [11-12], which offer the users great convenience.

As with the Android smart phones, the robots have gradually played an important role in all aspects of our daily life in recent years. For instance, the well known Roomba [13], an autonomous vacuum cleaner, has already been the world's best-selling robot. Others are like the multi-agent multi-sensor [14] based real-time sensory control system, the intelligent security robot fire detection system [15], the intelligent home security robot [16], the telepresence mobile robot for security applications [17] and the mobile robot tested with manipulator for security guard applications [18], which are convenient for our life. Thus, the combination of smart phones and mobile robot has been an inevitable trend.

OpenWrt [19] is described as a Linux distribution for embedded devices, which provides a fully writable file system with package management and allows users to customize the device through the use of packages to suit any application. Many types of devices support the OpenWrt operating system, such as network devices e.g. routers or Internet stations, embedded devices, mobile phones and so on. OpenWrt reliability has been used in several large projects related to communication in remote areas such as VillageTelco project [20], Wray Project in England [21], AirJaldi Network in India [22], Pebbles Valley Mesh Network in South Africa [23], Sengerema Mesh Network [24] and Tegola Mesh [25].

This paper proposes a system to build a smart car controlled by an Android application in two ways, and the video surveillance and temperature can be shown on the smart phones in real time. The OpenWrt wireless router acts as a transfer station of the whole system.

\*Corresponding Author: Quan-Ming Zhao  
(E-mail: qmzh@hebut.edu.cn)

<sup>1</sup>The school of Electronics and Information Engineering, Hebei University of Technology, No.5340, Xiping Road, Beichen District, Tianjin, China

## 2. System Structure

The system consists of two sections: an Android client and a mobile platform (as shown in Figure 1). Both of them are in the same segment and linked to the main router via WiFi. Wifi is a wireless communication technology with high security in long distance. It's also a specification to wirelessly transfer data based on IEEE 802.11. The highest transfer data rate is up to 560Mbps.

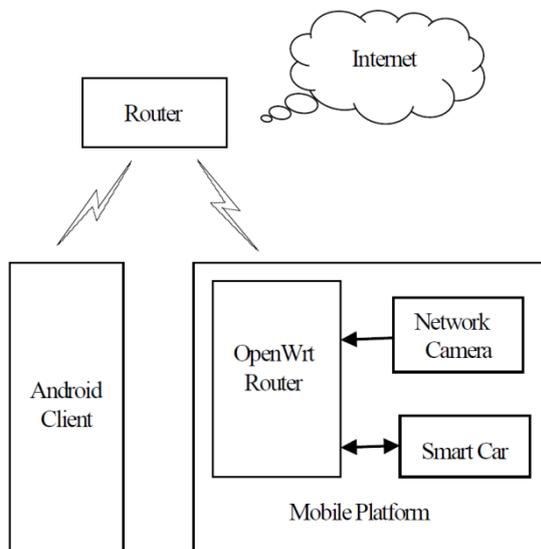


Figure 1: Overall design of the system

The mobile platform mainly contains a smart car based on STC89C52 MCU, a network camera and an OpenWrt wireless router in detail. The OpenWrt router is not only a wireless router, but also has a Linux system. Many softwares and applications can be installed conveniently to extend its functionality. The smart car is controlled by the Android client in two ways. One way is to use a gravity sensor of the Android phone to send commands to the car. The other way is to click the buttons designed in the main interface. The speed is adjustable. The temperature and video can be displayed on the Android phone once the socket connection is successful. The progress of the communication system can be described as follows.

The Android client can establish a socket connection with the OpenWrt router based on TCP or UDP. There are two forms of data to be transmitted. One is control instructions and data sensed by the temperature sensor, which can be transmitted through serial ports. It's a two-side serial communication. The other is a video stream. The Android client can access MJPEG-streamer which is a server installed in OpenWrt router and receive video stream via a URL. Acting as a transfer station of communication, the OpenWrt router transmits the data received with different ways independently. However, the OpenWrt

router is set to repeater mode to increase communication distance.

TCP offers reliability by providing connection-oriented, end-to-end, reliable packet delivery through a network. Generally it's used to ensure the reliability of long distance data transmission. UDP diagrams can be lost or delivered in a different orders to the receiving application. The transmitted data contains commands which need to be interpreted correctly, so TCP is more suitable for the system.

## 3. Hardware Realization

The hardware part is designed with a smart car, an OpenWrt wireless router and a network camera.

### 3.1 Smart Car

The smart car is assembled and built with several function modules as follows. STC89C52 is chosen as the main control chip, which has 8KB of flash memory storage and three timers. That's enough to support the system. Other modules include power modeled motor, L298N motor driver and DS18B20 temperature sensor (as shown in Figure 2).

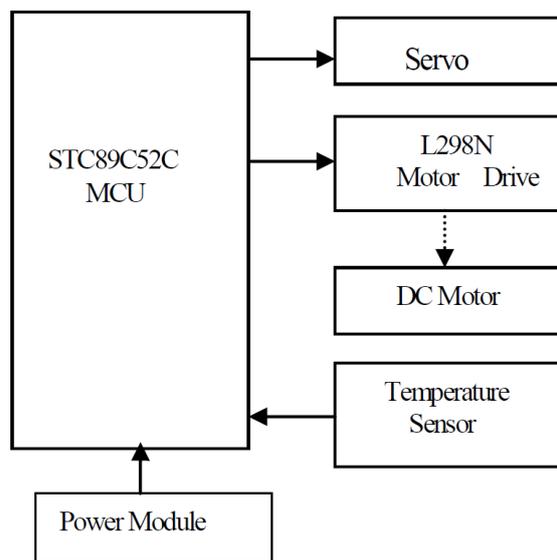


Figure 2: Components of the smart car

The car can complete a series of actions with the velocity difference between the left and right motors, including moving forward, backward, acceleration, deceleration and corner operation. For the motors, there are three problems to settle.

- 1) The current from I/O interface of MCU is not high enough to drive a motor which is a kind of high-power device.
- 2) Implement the function of commutation control.
- 3) Adjust the speed of the motors.

Generally, L298N module is used to solve the first two problems. L298N contains two H bridge circuits, and one H bridge circuit is required to control one motor. The third problem is resolved by the output PWM signals from the STC89C52. It has been concluded that the DC motor speed changes with the PWM duty cycle, which can be realized by a timer interrupt subroutine.

The servo is commonly controlled by a 20ms cycle PWM signal, whose high level at 0.5ms~2.5ms corresponds to 0~180 degrees linearly (as shown in Table 1). The car is equipped with two servos for the network camera to rotate freely from 0 to 180 degrees in vertical directions and horizontal directions.

**Table 1: Correspondence between high level time and degree**

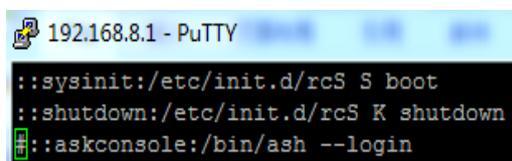
High level Time	Degree
0.5ms	0°
1.0ms	45°
1.5ms	90°
2.0ms	135°
2.5ms	180°

### 3.2 OpenWrt Router

GL-iNet 6416 is the mini wireless router chosen for the system. The mini wireless router takes AR9331 architecture, and supports 802.11 b/g/n. The interfaces include 1 WAN, 1 LAN, 1 USB2.0, 1 micro USB (power), and 1 Reset button. The transmission rate is 150Mbps. There are three steps to be an OpenWrt router from a common one.

- 1) Develop the firmware which can be found on the official website.
- 2) Install corresponding applications according to the function to be realized. The following applications need to be installed correctly: kmod-video-core (the kernel module of network camera), kmod-video-uvic (the camera driver), mjpg-streamer\_r182-1(a video stream server), and ser2net (serial to network).
- 3) Thirdly, configure parameters with the help of software-- putty.

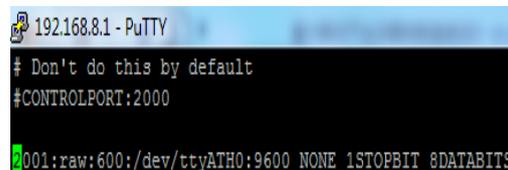
Login the OpenWrt system with putty, as shown in Figure 3, add an “#” ahead of the last line to support two-side serial communication.



**Figure 3: Setting of two-side serial communication**

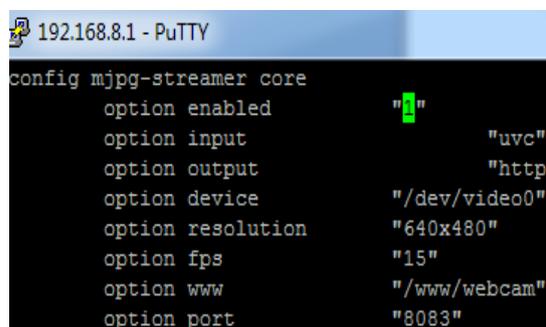
Figure 4 shows the configuration parameters of serial communication.

2001: Port Number  
 600: Time of a network connection /dev/ttyATH0:  
 the serial device  
 9600: Baud rate



**Figure 4: Set parameters of ser2net**

Figure 5 shows the configuration parameters of MJPG-stream. MJPG\_streamer is a free video stream server, which can be applied to the network based on TCP/IP. Images from a network camera can be transferred to the browsers in JPEG format. The MJPG\_streamer would return the video stream with HTTP response header when the client sends a request to it.



**Figure 5: Set parameters of MJPG-streamer**

Generally speaking, an URL contains the protocol type, host, number of port and resource Name. The format can be described as:

Protocol: //host:port/?action=resourceName.

8083 is the access port of video surveillance for android client. The URL should be:

<http://192.168.8.1:8083/?action=snapshot> or  
<http://192.168.8.1:8083/?action=stream>

The difference between the two URLs is that we can get a static image through the former, and a video stream from the later. For the action snapshot, the static image received should be refreshed constantly so that static images can be displayed on the phone at a high speed, which looks like playing a video. The data format of each image (in JPEG format) contains two sections: tag and compressed data. ‘FFD8’ is the beginning tag, and ‘FFD9’ is the ending tag. We need to get each frame of the video through interpreting the received content stored in buffer according to the data format.

### 3.3 Network Camera

MJPEG and YUV are the most common output video formats for cameras. Video data from cameras should be transferred into the JPEG format that is the supported formats of browsers. MJPG is a kind of compressing technique for dynamic images based on JPEG. MJPG-streamer would transfer the images in JPEG format to the server directly if the camera support MJPG output format. If not, the original image would be compressed into a JPEG file, which would consume CPU. Besides, the video may be not smooth at high resolutions in YUV format. Thus, the network camera should satisfy the following requirements for the system.

- 1) Support UVC driver.
- 2) Support MJPG output format.
- 3) Connect with the OpenWrt router through an USB interface.

### 4. Software Realization

The system is realized by two sections on software, MCU program and android application.

#### 4.1 Design of MCU Program

The MCU program is written in the C programming language and debugged through Keil u Version4. The flow chart of MCU program is illustrated in Figure 6.

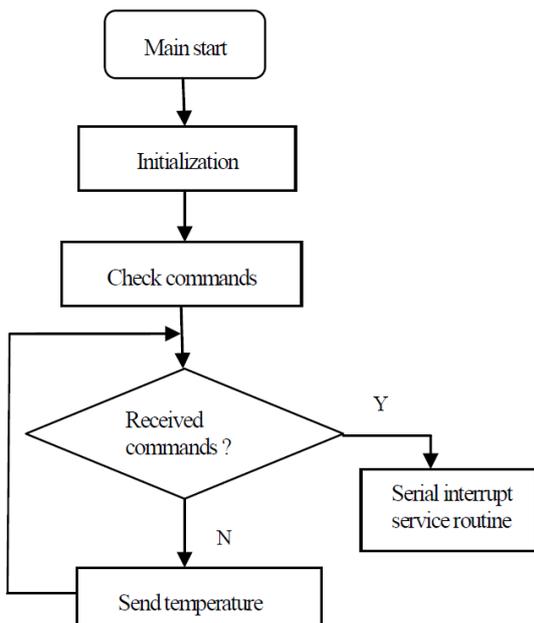


Figure 6: Flow chart of MCU program

The MCU program starts with the initialization function, which includes initializing the I/O, timers, the DS18B20 temperature sensor and serial ports.

In the main program, it needs to check circularly whether commands are received from the android client side. If not, the temperature sensed by DS18B20 will be sent back to the Android client to display every 4 seconds. Once received, the serial interrupt service routine would be executed (as shown in Figure 7). Each command is sent by the Android client as an array of 6 bytes.

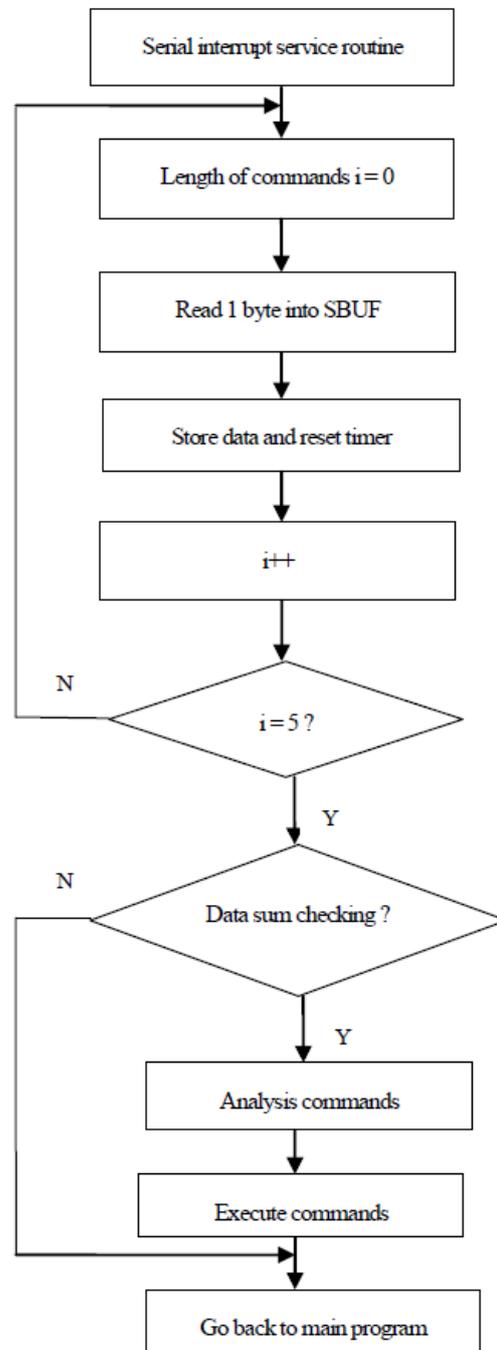


Figure 7: Flow chart of serial interrupt service routines

It's a two-side communication through serial ports. The parameters of serial communication are set to the following:

Baud: 9600

Verification: None

Data bits: 8

Stop bits: 1

### 4.2 Design of Android Application

Programs for the Android client are written in the JAVA programming language and developed with Eclipse. Figure 8 shows the flow chart of Android application. There are two tasks for the Android application. One is to send commands to control the car, and the other is to receive data, including temperatures and video streams.

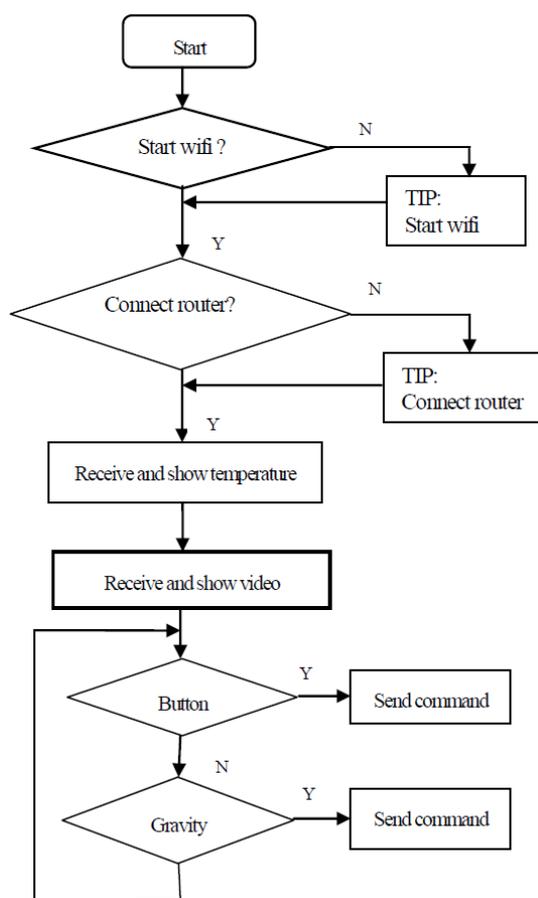


Figure 8: Flow chart of Android applications

Firstly, the task of receiving data will be illustrated. The time-consuming operation in the program would block the main thread; besides it's not allowed to update UI in the main thread, there are four child threads to accomplish specific functions. The child threads are listed as follows.

- 1) The WifiScanThread is to check the WiFi signal strength, and there would be a toast to prompt whether the phone has turned on WiFi or it is out of the WiFi coverage.

- 2) The TCPserverThread is to establish a socket connection between the OpenWrt router and the android phone. The progress can be described as follows. The OpenWrt router acting as the server side would listen to the user-defined ports all the time. The server would make a response to the client side by function outputstream once it has listened to a request for connection. All the specific functions can't be realized once this child thread fails to start.
- 3) The RecTemperatureThread is a thread to receive temperature data sent by MCU and update UI every 4 seconds.
- 4) The RecVideoThread is to play the video collected by the network camera once connected with the OpenWrt router.

Secondly, the car is controlled by the commands sent by the Android client in two ways. Each command would be transmitted to the OpenWrt router through WiFi.

One way is to use gravity sensors of the Android phone to send commands to the car. Gravity sensors measure the gravitational force applied on an object in all 3 axes. The coordinated system of sensors is different from that of the phone screen. The x axis points to the right of the screen, the y axis points upward along the screen and the z axis is perpendicular to the screen outward (as shown in Figure 9). The left and right movements are defined by the x-axis, and the upward and downward movements are defined by y-axis when the phone is set to a vertical screen mode. The gravity sensor is very sensitive even to a steady movement. The orientation change of the Android phone makes a difference in the value of force applied due to gravity. The value would be sent to the car side to control the car's movement.

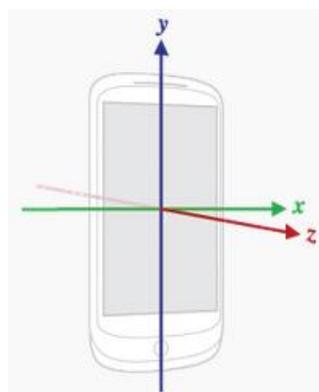


Figure 9: Coordinated system of Sensor

The other way is to click the buttons designed in the main interface. Each button represents a direction. The car would make a corresponding response in real time once the button is clicked. Besides, a seekbar is designed to fulfill the acceleration or deceleration.

### 5. Experiments and Results

To fulfill the proposed functions, ensure that both the Android phone and the OpenWrt router have been connected to the Internet with the same network before testing.

Firstly, set parameters in the setting interface, including the IP address of the OpenWrt router, the port which is listened by the OpenWrt router to establish a socket connection and the access port of video surveillance for the Android client. Both the ports must be the same with ones configured in the OpenWrt system respectively (as shown in Figure 10). The blue seekbar in the main interface is used to control the speed of the car.



Figure 2: Main Interface and Setting Screen

Figure 11 shows the screen shot of the video collected by the network camera. The video can be played smoothly once the Android phone is communicated with the OpenWrt router. Meanwhile, the temperature can also be displayed correctly and refreshed constantly.



Figure 11: Images of video surveillance

It's necessary to print the commands that the car received to check whether the car has run correctly. Each command is sent by the Android phone as a fixed-length array. The user-defined data format of communication protocol between the OpenWrt router and the android phone is described as follows (as shown in Table2). Data sum is adopted in transmission to avoid the communication errors. The formula is: Check bit= Type+Data1+Data2.

Table 2: Data format of communication protocol

Head	Type	Data1	Data2	Checkbit	End
FF	XX	XX	XX	XX	FE

For example, the command for moving forward is defined as 'FF 00 01 00 01 FE'. The command which the car received has been printed on the computer with the help of a USB to serial line (as shown in Figure 12).

The parameters of serial communication are set to the following:  
 Baud: 9600  
 Verification: None  
 Data bits: 8  
 Stop bits: 1



Figure 12: Command received for moving forward

Other commands can also be tested in the same way.

## 6. Conclusion

Designed is a remote-control system based on a smart car for indoor environment. The smart car controlled by the Android phone can move around the room freely. Meanwhile, the video surveillance and the temperature of the room can be displayed on the phone. Besides, the phone can be shackled to certain directions to control the car with gravity sensor. The proposed functions have been completed.

There is also much to be done to improve the functions. For example, the OpenWrt router and the Android phone both are connected to the same network. For the different network segment, it can be realized with NAT (Network Address Translation).

## Acknowledgments

This work was supported by the Fund No. 15227408D of Hebei Province Science and Technology Plan and the company Idriver Plus.

## References

- [1]. Chen H, Wan L, and Wang F, "Model predictive controller design for the dynamic positioning system of a semi-submersible platform," *Journal of Marine Science and Application*, 11(3) 2012, pp. 361-367.
- [2]. Z. Luo, S. Han, and F. Duan, "The development of a smart house system based on Brain-Computer Interface," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 1012-1017.
- [3]. C. Kidd, R. Orr, G. Abowd, et al. "The aware home: A living laboratory for ubiquitous computing research," *Cooperative buildings, Integrating information, organizations, and architecture*, 1999, pp. 191-198.
- [4]. T. Yamazaki, "Ubiquitous home: real-life testbed for home context-aware service," *International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2005. Tridentcom. IEEE, 2005, pp. 54-59.
- [5]. C.Giovanni, D.Berardina, and P.Sebastiano, "Social robots as mediators between users and smart environments," in *Proceedings of the 12th international conference on Intelligent user interfaces*, New York, NY, USA, 2007, IUI '07, pp. 353-356, ACM.
- [6]. Baeg S H, Park J H, Koh J, et al. "Building a smart home environment for service robots based on RFID and sensor networks," in *Control, Automation and Systems*, 2007. ICCAS '07. International Conference on, Oct. 2007, pp. 1078-1082.
- [7]. M. Aiello, "The role of web services at home," *International Conference on Internet and Web Applications and Services/advanced International Conference on Telecommunications*, 2006. Aict-Iciw. 2006, pp.164.
- [8]. J. Parra, M.A. Hossain, A. Uribarren, et al. "Flexible smart home architecture using device profile for web services: a peer-to-peer approach," *International Journal of Smart Home*, vol. 3, no. 2, 2009, pp. 39-56.
- [9]. Zhao Wang, Eng Gee Lim, Weiwei Wang, et al. "Design of an arduino-based smart car," *SoC Design Conference (ISOCC)*, 2014 International. IEEE, 2014, pp. 175-176.
- [10]. J.Brodtkin, "On its 5th birthday, 5 things we love about Android," Available from: <http://arstechnica.com/gadgets/2012111lon-androids-5th-birthday-5-things-we-love-about-android/>, 2013.
- [11]. Rolando B C, Julio B O, and Esaías P A, "Controlling Digital Dimmer Through Mobile Phone," *Electronics, Communications and Computer (CONIELECOMP)*, 2010 20th International Conference on, 22-24 Feb, 2010, pp.57-61.
- [12]. Zhang C, Qin H, and Qu L, "LED Dimming System of Wireless Remote Controller," *Image Analysis and Signal Processing (IASP)*, 2012 International Conference on, 9-11 Nov. 2012, pp.1-4.

- [13]. J.L. Jones, "Robots at the tipping point: the road to iRobot Roomba," IEEE Robotics & Automation Magazine, 13 (1) 2006, pp. 76–78.
- [14]. R. C. Luo, and K.L. Su, "A multiagent multisensor based real-time sensory control system for intelligent security robot," Proceedings on ICRA '03. IEEE International Conference on Robotics and Automation, Vol. 2, 2003, pp. 2394 -2399.
- [15]. R. C. Luo, K.L. Su, and K.H. Tsai, "Intelligent security robot fire detection system using adaptive sensory fusion method," IEEE 2002 28th Annual Conference of the IECON 02 Industrial Electronics Society, Vol. 4, 2002, pp. 2663 -2668.
- [16]. R.C. Luo, T.Y. Hsu, T.Y. Lin, et al. "The development of intelligent home security robot," Proceedings of the 2005 IEEE International Conference on Mechatronics, 2005, pp. 422-427.
- [17]. R.J. Schultz, R. Nakajima, and J. Nomura, "Telepresence mobile robot for security applications," Proceedings of Industrial Electronic, Control and Instrumentation (IECON'91), Vol.2, 1991, pp. 1063-1066.
- [18]. M. Saitoh, Y. Takahashi, A. Sankaranarayanan, et al. "A mobile robot tested with manipulator for security guard application," Proceedings on 1995 IEEE International Conference on Robotics and Automation, Vol. 3, 1995, pp. 2518-2523.
- [19]. OpenWRT website: <http://openwrt.org/>.
- [20]. Adeyeye M and Gardner-Stephen P, "The Village Telco project: a reliable and practical wireless mesh telephony infrastructure," EURASIP Journal on Wireless Communications and Networking 2011.1 pp. 1-11.
- [21]. Ishmael J, Bury S, Pezaros D, et al. "Deploying rural community wireless mesh networks," Internet Computing, IEEE 12.4, 2008, pp. 22-29.
- [22]. Surana S, Patra R, Nedeveschi S, et al. "Beyond Pilots: Keeping Rural Wireless Networks Alive," NSDI. Vol. 8. 2008, pp. 119-132.
- [23]. Johnson, David Lloyd, and Kobus Roux, "Building rural wireless networks: Lessons learnt and future directions," Association for Computing Machinery, 2008, pp. 17-22.
- [24]. Lenoir M, "Building a Wireless Mesh Network in Sengerema, Tanzania," (2008) <http://www.iicd.org/articles/launch-of-wireless-internet-in-sengerematanzania> (accessed by 15 January 2015).
- [25]. Bernardi G, Buneman P, and Marina M K, "Tegola tiered mesh network testbed in rural Scotland," ACM Workshop on Wireless Networks and Systems for Developing Regions, San Francisco, California, USA, Sep, 2008, pp. 9-16.



**Quan-ming Zhao** received Bachelor in Radio Technology from Hebei Institute of Technology in 1991, Ph. D in Signal and Information Processing from Tianjin University in 2007. He is currently a professor of Hebei

University of Technology and his research interest covers electromagnetic field and microwave circuit design.



**Pei-pei Jin** is studying the Master Degree in the school of Electronics and Information Engineering in Hebei University of Technology, majored in communication and electronic measurement technology.