Development of Fire Alarm System using Raspberry Pi and Arduino Uno

Conference Paper · December 2013

1 author:

Norlida Buniyamin
Universiti Teknologi MARA

Development of Fire Alarm System using Raspberry Pi and Arduino Uno

Md Saifudaullah Bin Bahrudin
Faculty of Electrical Engineering
University Teknologi MARA
Selangor, Malaysia
c_pool07@yahoo.com

Rosni Abu Kassim
Norlida Buniyamin
Faculty of Electrical Engineering
University Teknologi MARA
Selangor, Malaysia

Abstract— **The proposed Fire alarm system is a real-time monitoring system that detects the presence of smoke in the air due to fire and captures images via a camera installed inside a room when a fire occurs. The embedded systems used to develop this fire alarm system are Raspberry Pi and Arduino Uno. The key feature of the system is the ability to remotely send an alert when a fire is detected. When the presence of smoke is detected, the system will display an image of the room state in a webpage. The system will need the user confirmation to report the event to the Firefighter using Short Message Service (SMS). The advantage of using this system is it will reduce the possibility of false alert reported to the Firefighter. The camera will only capture an image, so this system will consume a little storage and power.**

Keywords-component; Raspberry Pi; Arduino Uno; fire; alarm; webserver

I. INTRODUCTION

Fire is an undesirable event that could bring a great loss of social wealth and human life. To prevent this losses, various alarm systems have been developed such as smoke detectors, temperature sensor based systems etc. As technologies evolved and instruments such as temperature sensors, camera etc becomes affordable, various automated fire alarm systems are now available. In conjunction with the cheaper instruments, internet based and wireless broadband technologies, have also improved and there are now various systems that enables cheap, high rate data transmission and wireless networking. The availability of cheap credit card sized single board computer such as the Raspberry Pi has enabled the creation of numerous automated and monitoring system that has low power consumption, faster processing ability at a lower cost. The fire alarm system proposed in this paper integrates the use of affordable instruments, connectivity and wireless communication.

A. Background study and overview

Various research and numerous prototypes of automated security surveillance system have been developed using various platforms. Recent improvements in GSM, web server and microcontroller technologies have led to the developments of various fire alarm systems. For example, Yu Qiongfang et.al [1] proposed an intelligent fire alarm system using fuzzy neural network. It process data from the sensors and calculates model of fuzzy neural network based on the characteristics of fire detection signal so it will has a self-learning and adaptive capability. However, it is only a study on proposing a new approach for fire detection.

Cao Shunxia et.al [2] designed a wireless intelligent home alarm system consisting on anti-theft feature, anti-fire feature, and anti-harmful gas leak feature using Single Chip Microcomputer (SCM) AT89C51 and voice chip ISD1420. Two SCMs were used to display the gas concentration and alarm host as the alarm signals were sent by using wireless transmission. When the sensor detects smoke, a voice message will be sent to the relevant department. However, if an error occurs during the detection, a false alarm will be submitted because this system did not include any user confirmation.

Jun Hou et.al [3] proposed an intelligent home security system using Zigbee to monitor important locations inside a home through a surveillance camera. When the system was triggered by any penetration, the user will be notified through SMS and Multimedia Message Service (MMS). The temperature and gas sensor were connected to the system motherboard using Zigbee modules and forming a Wireless Sensor Network (WSN). Even though it can be included as one of the most advanced system, the system motherboard used to manage the WSN was too expensive.

Rakesh V S et.al [4] improved real-time surveillance system for home security system using Beagleboard SBC, Zigbee and FTP Webserver which monitor important locations inside a house using camera and detecting smoke. When smoke or intruder movement is detected, the system sends warning messages through SMS to cell phones, starts capturing real-time video for fixed duration and makes the alarm on. But this system only sends SMS to warn the user and cannot broadcast the live streaming video as the system record the video only. Moreover, the single board computer used is expensive and has lower technical specifications compare to the Raspberry Pi.

The solution for the problems from the previous research as stated above is to develop a new fire alarm system that alerts the user instantly when any events occur and asks for permission from the user to report to the Firefighter. The purpose of this study is to implement a fire alarm system using a cheaper single-board computer, the Raspberry Pi, combined with a microcontroller board, the Arduino Uno and the use of mid-level and high level programming languages to write the program. The project is a Bachelor degree final semester project that has the objective to enable the student to apply knowledge gained from the previous semesters.

II. SYSTEM ARCHITECTURE

A. Selecting a Template (Heading 2)

Figure 1 shows the system architecture. The structure of this fire alarm system is composed of five components, which are Raspberry Pi Model-B single-board computer, Arduino Uno single-board microcontroller, gas sensor, GSM shield and webcam. Raspberry Pi was selected due to its good technical specifications, high performance for data processing and is cheaper than other single board computers available in the market. The Arduino Uno is used to reduce the tasks of the Raspberry Pi by processing the analog signal from sensor and send SMS using the GSM shield. The webcam is used to capture the image of the environment when the sensor detects any abnormality in the air.

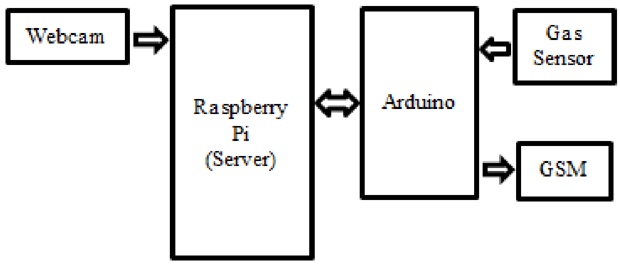

Fig. 1: Block diagram of the fire alarm system

III. METHODOLOGY

A. Raspberry Pi

Raspberry Pi project development started on 2006. It is an inexpensive computer that uses Linux-based operating system, equipped with a 700 MHz ARM-architecture CPU, having a 512 MB RAM and featured with two USB ports and an Ethernet controller [5]. It is able to handle full HD 1080 video playback and by using the onboard Videocore IV graphics processing unit (GPU), it was able to demonstrate the ported version of the Quake 3 game. It allows flexibility in the choice of programming languages and installation of software that could be used [6]. It has the ability to serve webpage by installing Apache HTTP Webserver on it [7]. Arduino Uno with GSM Shield.

Arduino Uno is an ATmega328-based microcontroller operated using 5.5 Volts input voltage supply and has a maximum operating frequency of 20 MHz [8]. The power supply for the Arduino Uno can be either from a USB connection, DC power supply, or both. It is a high performance device which has low power AVR 8-bit Microcontroller with 32K bytes in-system and advanced reduced instruction set computing [9].

The GSM shield controller used is SIM900 and it was attached to the Arduino Uno. Its function is to send SMS to the Firefighter. The SIM900 GSM shield has low power consumption. It has its own GSM communication module and processor which is programmed using JAVA [10]. It can be used to send and receive SMS, voice calling and receive data packet. The AT commands is used to enable the GSM shield to send the SMS to the recipient.

B. Sensor

The sensor used in this system is gas sensor QM-NG1 to detect the presence of smoke. It can be placed at any locations inside the room, but it is best placed at the center of room. The selection of the sensor was made due to its high sensitivity for poisonous gas, long inductive time and long life span [11].

C. Hardware connections

The USB webcam is connected to the Raspberry Pi by using an USB cable and the GSM shield was attached to the Arduino Uno. A Local Area Connection cable is plugged in to the Ethernet port of Raspberry Pi to allow a connection to the internet. The ports of Raspberry Pi used to connect with the Arduino Uno are shown in Fig. 2a including the pins of the Arduino Uno. Port 24 of Raspberry Pi is connected to Pin 13 of Arduino Uno, Port 25 is connected to Pin 12, and Port 23 is connected to Pin 11. Port 24 is used by the Raspberry Pi to get warnings from Arduino Uno if any events occur. Port 25 is used to inform the Arduino Uno if the user confirmed the alert, while Port 23 is used to inform the Arduino Uno if the user cancelled the alert.

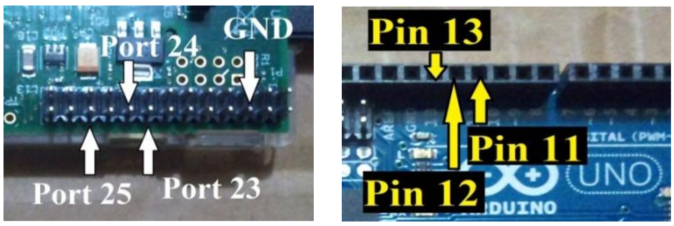

Fig. 2: The ports of Raspberry Pi and the pins of Arduino Uno

However, the output voltage from the digital output pins of Arduino Uno is 5V, while the input ports of Raspberry Pi can only handle 3.3V. Any input voltage higher than 3.3V applied to the input ports of Raspberry will permanently damage the Raspberry Pi [12], so a voltage divider circuit was designed to reduce the output voltage from the Arduino Uno. The calculation for the voltage divider circuit is shown below and the image of the voltage divider circuit is shown in Fig. 3:

$$Vout = \frac{R2}{R1 + R2} \cdot Vin$$

$$Vout = \frac{10K}{5K + 10K} \cdot 5V$$

$$Vout = 3.3V$$

The analog output signal from the sensor will be processed by the Arduino Uno. The sensor got the supply voltage from Arduino Uno 5V pin, and the ground pin of the sensor was connected to ground pin of Arduino Uno. The analog output signal pin of the sensor was connected to analog input pin of Arduino Uno. Fig. 3 shows the connection between the sensor and Arduino Uno.

Fig. 3: The Voltage Divider Circuit and connections between Arduino Uno and the sensor

A monitor, an USB keyboard and an USB mouse were used for developing the software in the Raspberry Pi. The block diagram of the system is shown in Fig. 4. The monitor is connected to the HDMI port of Raspberry Pi using a HDMI cable, while the keyboard and the mouse were connected to the USB ports of the Raspberry Pi.

Fig. 4: Full block diagram of the system

D. Software

The Raspberry Pi used a Linux-based operating system to operate. In this project, Raspbian Wheezy operating system (OS) was chosen because it is recommended by the manufacturer. The Raspbian Wheezy is the modified version of the Debian Cheezy OS. The OS image was extracted to an SD card by using the Win32 Disk Manager. The configurations of the OS such as login details and IP address settings were done. The login at startup was disabled to make sure that if the power supply to the Raspberry Pi was cut down, the Raspberry Pi will never ask for login details. The Apache HTTP webserver was installed to make the Raspberry Pi as a webserver. The development of the programming in the Raspberry Pi is done by using the IDLE software. High level programming language selected for writing the OS application program for the system is Python programming language. The IDLE software is the Integrated Development Environment (IDE) for Python and it was already installed in the OS. A SimpleCV library for Python was installed to enable the operation of capturing the image done by the webcam.

The Arduino Uno used open-source Arduino Uno integrated development environment (IDE) for compiling and uploading programs to the board. The Arduino Uno programs can be written using C or C++ language. These two languages were categorized as an intermediate-level language [13]. The Arduino Uno is used to monitor the signal output from the sensor.

To enable the user from outside of the local network to access the webpage, it must be visible in the internet. Port forwarding technique was used to allow the public devices to access the internal server that has a private IP address [12]. Two ports were port forwarded, which are Port 80 for HTTP port and Port 443 for Apache. Then, a hostname was registered www.no-ip.com and linked to the Wide Area Network (WAN) IP address. The registered hostname is http://saifubahrudin.no-ip.biz. To keep the current IP address in sync with the No-IP host, a client was installed to continually checks and automatically updates the Domain Name Services (DNS) if the IP address ever changed.

IV. SYSTEM DESCRIPTION

The proposed fire alarm system that used Raspberry Pi as a master device and Arduino Uno as a slave device was implemented. The sensor was placed at the center of the room. It is sensitive to the ionization of the surrounding air, resulting in the changes of its resistivity. The value of the output voltage from the sensor circuitry varies linearly with the resistivity of the sensor, so any changes in the sensor resistivity will result in the changes of the output voltage. The output voltage of the sensor circuitry is read in 1024 bit resolution by the Arduino Uno. At start, the Arduino Uno read the first reading from the sensor and set the threshold value. The calculation of the threshold value is shown below:

$$Threshold = 2.5 \times sample\ reading$$

As for the ports configurations of the Raspberry Pi, Port 24 was defined as the input port to receive any alert from Arduino Uno, Port 25 was defined as the output port and it is used to inform the Arduino Uno if the user has already confirmed the alert, while Port 23 was defined as the output port and is used to inform the Arduino Uno if the user cancelled the alert. While for the pins configurations of the Arduino Uno, Pin 13 was defined as the output pin for sending the alert to Raspberry Pi, Pin 12 and Pin 11 were defined as the input pin for receiving the information from the Raspberry Pi whether the user confirmed or cancelled the alert.

The Raspberry Pi waits for any input from the Arduino Uno on its Port 24. Upon detecting the event, the Arduino Uno will send a HIGH output which is 5V from its Pin 13 to the Ports 24 of the Raspberry Pi and the voltage will be reduced as the current flows through the voltage divider circuit. An image will be captured using the webcam. The image will be displayed on the webpage and the Raspberry Pi will wait for the user confirmation on the validity of the alert. The user can do the confirmation by clicking a submit button on the webpage. At the moment the user click, the PHP programming of the webpage will automatically open a text file named Logger file and log a command data. The Python programming will open the text file, and write a word "send" in the Logger_file. Then the Raspberry Pi will send a HIGH output which is 3.3V through Port 25 to Pin 12 of the Arduino Uno. The Arduino Uno will then generate an SMS alert and send it using the GSM shield.

If the user cancels the alert, the PHP programming of the webpage write a word "cancel" in the Logger_file. The Python programming read the content of the Logger_file, then the image will be deleted and the Raspberry Pi will send a HIGH output through Port 23 to Pin 11 of the Arduino Uno. The Arduino Uno will continue to read the analog signal from the sensor and the Raspberry Pi wait for other alert from the Arduino Uno. The flowcharts for the programming in the Arduino Uno and the Raspberry Pi are shown in Fig. 5 and Fig. 6.

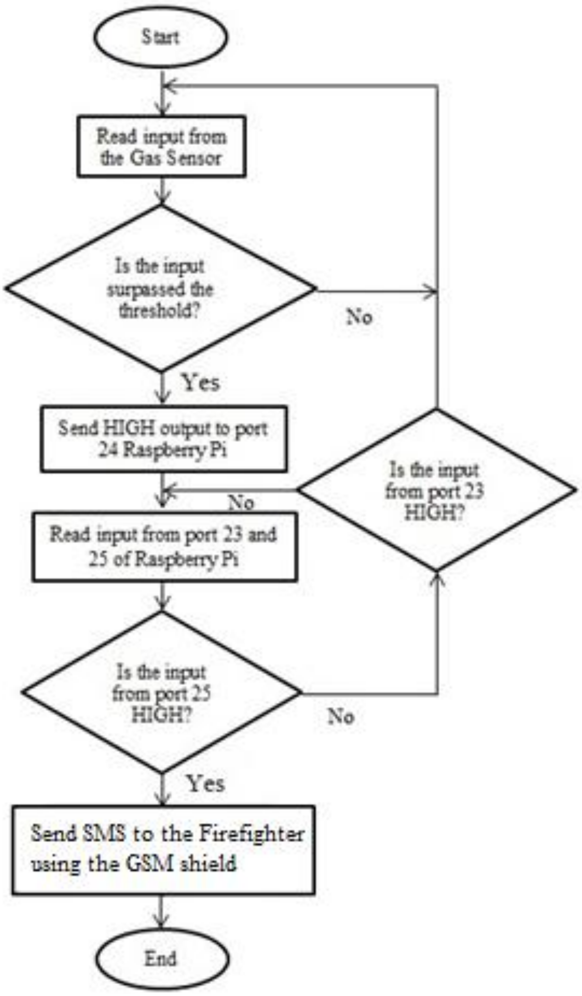

Fig. 5: The flowchart of the C programming in Arduino Uno

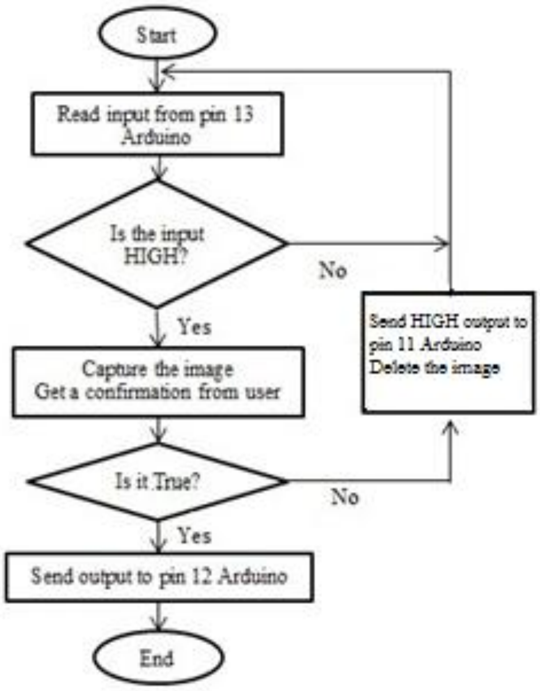

Fig. 6: The flowchart of the Python programming in Raspberry Pi

V. RESULT AND DISCUSSION

The system was tested in two different situations; in the first test, the sensor was placed inside a room and some rubbish was burnt in the room to fill it with smoke and in the second test, the sensitivity of the sensor was changed after the sample reading for threshold value was taken. The first test is defined as True Warning while the second test is False Warning. The result will be displayed in the webpage. The webpage can be accessed at http://saifubahrudin.no-ip.biz.

At first, the user will be prompted to fill the login details as shown in Fig. 7. After logged on, the user will be redirected to the display webpage as shown in Fig. 8, which will display the image captured later.

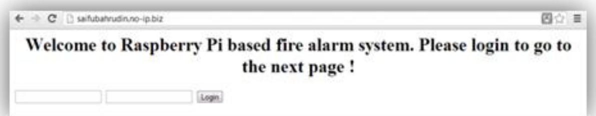

Fig. 7: The login page

Fig. 8: The default webpage

A. *True warning*

As the sensor detects the increase of coal gas in the air and the value of the measurements surpassed the threshold value, the camera captured an image and it was displayed on the webpage. Fig. 9 shows the display of the image in the webpage. After the submit button was clicked, the PHP programming modified the content of the Logger_file by writing a word “send” in it. The user will be redirected to the report webpage as shown in the Fig. 10. The Python programming read the content of the Logger_file and a HIGH output was sent through Port 25 of the Raspberry Pi. The Arduino Uno reads the signal as a confirmation for sending SMS to the Firefighter. The Arduino Uno activates the GSM shield and sent the address of the scene in the form of SMS by using AT Commands. Fig. 11 shows the screenshot of the message sent by the Arduino Uno.

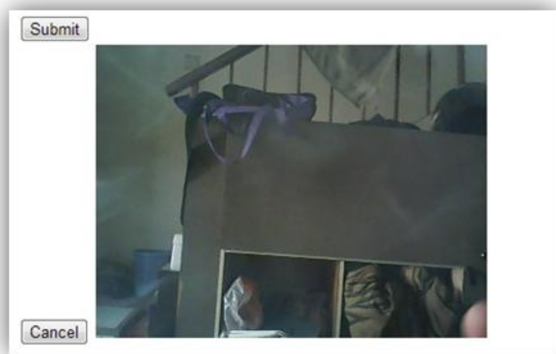

Fig. 9: The image displayed on the webpage.

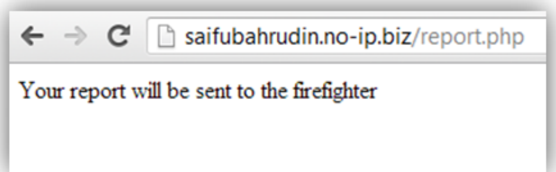

Fig. 10: The webpage redirected from the display webpage

Fig. 11: The screenshot of the message sent by Arduino Uno

B. False warning

A false warning test was made to show what happened when any error occurs during the operation of the system. The source of error can vary. It could be due to sensor failure or gas leakage. However, in this project, gas leakage detection was not covered, so if the sensor detects the increase of methane gas, it will be assumed as device error. It is dangerous to leak the methane gas in the room, so the sensitivity of the sensor was changed after a sample reading to determine the threshold value was taken. After the values of the measurements exceed the threshold value, an image was captured. The image was displayed on the webpage as shown in Fig. 12. After the cancel button was clicked, the PHP programming modified the content of the Logger_file by writing a word “cancel” in it. The user will be redirected to the cancel webpage named cancel.php as shown in the Fig. 13. The Python programming read the content of the Logger_file and a HIGH output was sent through Port 23 of the Raspberry Pi. The Python programming deleted the image and the content in the Logger_file. The Arduino Uno reads the signal from the Raspberry Pi as a cancellation and continued to monitor the measurements from the gas sensor. After ten seconds, the user will be redirected back to the default webpage.

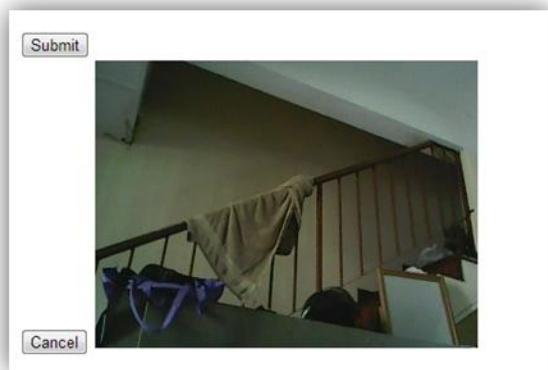

Fig. 12: The false warning image displayed on the webpage

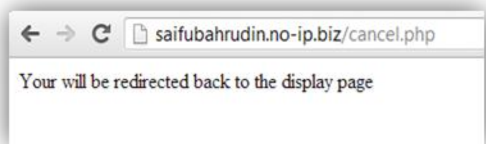

Fig. 13: The cancel webpage

VI. CONCLUSION

This study presented the development of a fire alarm system using the Raspberry Pi and Arduino Uno microcontroller. The developed prototype offered a feature that enabled verification that a fire actually occurred. The fire alarm system warns the user by first sending an alert and asks for confirmation before submitting a report/alert to the “Fire-fighter”. This system used low cost, reliable instruments that were suitable to develop a fire alarm and affordable as a final year project that enabled students to apply knowledge acquired during the engineering program.

VII. ACKNOWLEDGEMENT

The authors are grateful to Universiti Teknologi MARA for the use of the equipments and Laboratories.

VIII. REFERENCES

[1] Y. Qiongfang, Z. Dezhong, F. Yongli and D. Aihua, "Intelligent Fire Alarm System Based on Fuzzy Neural Network," *2009 International Workshop on Intelligent Systems and Applications*, p. 1, May 2009.

[2] C. Shunxia and C. Yanda, "Design Of Wireless Intelligent Home Alarm System," *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, p. 1511, August 2012.

[3] J. Hou, C. Wu, Z. Yuan, J. Tan, Q. Wang and Y. Zhou, "Research of Intelligent Home Security Surveillance System Based on ZigBee," *International Symposium on Intelligent Information Technology Application Workshops*, pp. 554-557, 2008.

[4] V. Rakesh, P. Sreesh and S. N. George, "Improved real-time surveillance system for home security system using Beagleboard SBC, Zigbee and FTP Webserver," *2012 Annual IEEE India Conference (INDICON)*, pp. 1240 - 1244, 2012.

[5] L. Garber, "UK Teams Build Supercomputer For $4050," *IEEE Intelligent Systems*, p. 19, November 2012.

[6] Jaseman, "The Pioneers," *The MagPi*, pp. 7-8, May 2012.

[7] B. Schaefer, "PiGauge," *The MagPi*, p. 14, December 2012.

[8] A. R. Krishna, G. S. Bala, A. Sastry, B. B. Sarma and G. S. Alla, "Design And Implementation Of A Robotic Arm Based On Haptic Technology," *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, no. 3, pp. 3098-3103, 2012.

[9] A. Abdullah, O. Sidek, N. A. Amran, U. N. Za'bah, F. Nikmat, H. Jafar and M. A. Hadi, "Development of Wireless Sensor Network for Monitoring," *2012 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, p. 109, 2012.

[10] S. Monk, Programming Arduino Getting Start With Sketches, McGraw-Hill, 2011.

[11] Ningbo Hi-Tech Park SKKY Science Technology Co. Ltd., QM-NG1, Zhejiang, 2010.

[12] Shawn and M. Richardson, Getting Started With Raspberry Pi, Sebastopol: O'Reilly Media, Inc, 2012.

[13] H. Schildt, C++: The Complete Reference, Mcgraw-Hill Osborne Media, 2003.

[14] L. Chun-yuan, "Design of intelligent fire alarm system based on GSM network," *2011 International Conference on Electronics and Optoelectronics (ICEOE)*, vol. 1, pp. VI-393, July 2011.

[15] B. Lee and J. Jeon, "An Embedded Router for Internet Communication Among Private Networks," *ICIT 2006. IEEE International Conference on Industrial Technology, 2006.*, p. 688, 2006.