위험지역 감시스마트로봇의 설계와 동작에 관한 연구

구경완 · 백동현*[†]

호서대학교 국방과학기술학과, *가천대학교 소방방재공학과

A Study of Smart Robot Architecture and Movement for Observation of Dangerous Region

Kyung-Wan Koo · Dong-Hyun Baek*[†]

Dept. of Defence Science Technology, Hoseo University

*Dept. of Fire Protection Engineering, Gachon University

(Received December 11, 2013; Revised December 26, 2013; Accepted December 26, 2013)

요 약

최근 원자력 발전소의 방사능 누출 사고, 삼성 불산 유출사건 등의 재난과 같은 사고는 사람이 직접 복구 작업이나 관리하기 매우 위험하다. 따라서 사람을 대신해 사고장소의 정찰이나 작업을 할 수 있는 무인로봇의 활용이 필요하며, 이를 위해 Mbed (ARM 프로세서)에서 온도/가스 센서와 습도 센서를 이용해 웹 카메라를 통해 영상을 전송받을 수 있도록 하여 시설물 곳곳을 감시할 수 있는 감시스마트로봇을 설계 시험하였다. 또한 외부에서도 인터넷망만 접속되면 HTTP Server를 통하여 PC, Android로 원격조종을 할 수 있도록 하여 시간, 장소에 관계없이 로봇을 원하는 곳으로 주행할 수 있도록 하였다. 이와 같은 감시스마트로봇은 24시간 감시 할 수 있으므로 사고발생의 최소화 및 범죄예방 등의 여러 목적으로 활용할 수 있고 사고 발생 시 빠른 대응을 할 수 있다. 아울러 사람대신 사용할 수 있으므로 인건비를 절감하여 경제적인 효과까지 기대할 수 있다.

ABSTRACT

Catastrophic disasters are sprouting out recently, i.e., the radiation leaks and the hydrofluoric acid gas leaks, etc. The restoration work for these kinds of disasters is very harmful and dangerous for human beings to handle themselves, thus allowing manless robots to fly the reconnaissance planes over to the disaster stricken areas and do the necessary work instead. For this endeavor and purpose, we created and tested an intelligent robot that can inspect those areas, using Mbed (ARM processor) technology temperature sensors and gas sensors aided by CAM (Computer-Aided Manufacturing) cameras. Also, HTTP Server, PC, androids and their combined efforts allow their remote controlled operation from far away with timing control. These intelligent robots can be on duty for 24 hours, minimizing the accidents and crimes and what not, and can respond more quickly when these misfortunes actually happen. We can anticipate the economic effects as well, derived from the reduced needs for hiring human resources.

Keywords : Mbed, Gas/Temperature/Humidity sensor, HTTP server, Remote control

1. 서 론

소득증가와 교통문화의 발달은 다양한 여가생활을 할 수 있게 하여 집을 비우는 일도 많지만 공장 같은 곳은 사람이 관리하지 못해 사고가 발생되기도 한다. 예로 삼성 불산 누출 사건도 불산이 누출되는 것을 감지하지 못하여 대처가 늦어졌기 때문이다. 더욱 무인 경비지역이거나 사람이 출입할 수 없는 지역에서는 빠른 대처가 필요하므로 현장을 감시하며 모니터링하여, 이상 징후 발견 시 상황을

알려주어 신속한 대처로 감시 작업의 효율성을 높일 필요가 있다. 이를 위해서는 여러 분야에서 인간의 필요에 따라 사람을 대신하여 작업을 수행 할 수 있는 로봇⁽¹⁻⁵⁾이 유용하다. 초기의 무인경비 시스템은 센서기반으로 인지된 상황을 실시간으로 관리자가 인지할 수 있도록 되어 있었다. 이는 많은 관리자와 항상 실시간으로 시스템을 관리해야 한다는 점에서 볼 때 매우 비효율적으로 사람이 직접 현장에 투입되지 않아도 된다는 사실 이외에는 큰 장점은 없었다. 그러나 IT기술과 카메라 기술의 발달로 인하여 감

[†]Corresponding Author, E-Mail: dhbaek@gachon.ac.kr
TEL: +82-10-3272-0118, FAX: +82-31-750-8749

ISSN: 1738-7167
DOI: <http://dx.doi.org/10.7731/KIFSE.2013.27.6.083>

시자가 24시간 시스템 옆이나 상황이 발생한 현장에 있지 않아도 인지 할 수 있도록 하는 시스템까지 연구되고 있다. 그러나 이러한 시스템은 대부분 부착형이거나 고착형으로 감시하고자 하는 지역의 구조나 넓이에 따라 매우 많은 센서와 감시 카메라, 이를 운용하는 센터의 복잡성이 함께 증가되는 형태였다.

따라서 본 논문에서는 감시지역의 복잡도나 넓이에 상관없이 이동형으로 상황을 인지할 수 있는 감시스마트로봇 시스템을 제안하고자 한다. 제안한 감시스마트로봇시스템은 CCD 카메라를 부착하여 가스, 온도, 습도 센서 이외에 직접 상황을 주시하고, 효율적으로 사건 발생 지역을 감시 및 관찰 할 수 있도록 하였다. 또한 감시로봇시스템과 이를 스마트통신 기능을 탑재한 Android 모바일 통신 기기와 연동하여 원거리에서도 상황을 판단하고 주시할 수 있게 하였으며, 센서기반으로 반응하는 현장상황의 문제점을 시각적으로 확인하여 상황 처리를 할 수 있도록 하였다.

2. 감시스마트로봇시스템

2.1 시스템 구현 주요 기능

감시스마트로봇시스템의 주요기능은 다음과 같다. 첫째, 사람이 접근하기 힘든 환경에서도, 온도나 가스분포도 상태를 지속적으로 모니터링 할 수 있으며 침입자의 유무를 판단하는 방법기능을 갖게 한다. 둘째, 현장을 어디에서나 카메라를 통해 감시할 수 있으며, 실시간 영상을 Android 기기나 PC로 전송할 수 있게 한다. 셋째, 특정 지역을 원하는 대로 감시할 수 있도록 감시스마트로봇의 조종을 Android 기기로 가능하게 한다. 넷째, 감시스마트로봇에 탑재된 온·습도 센서나 가스감지 센서 등을 통해 그 지역의 상태를 지속적으로 체크한다. 다섯째, 감시스마트로봇의 조종과 카메라 모니터링은 어디서나 간편하게 할 수 있도록 한다. Figure 1은 제안한 시스템의 블록 다이어그램이다.

2.1.1 PC 및 Android

PC는 인터넷 통신이 가능하며, 웹 페이지에 접속하여 로봇을 제어하고 데이터를 읽을 수 있다. 웹 서버를 사용하기

Figure 1. Block diagram of system.

때문에 OS의 제약이 적다는 점이 큰 장점이다. Android는 스마트 기능을 갖는 단말기를 의미하며 스마트 태블릿으로도 Wi-Fi만 된다면 PC와 똑같이 로봇을 조종할 수 있다. Android용으로 제작했으나 iOS에서도 호환이 되는 HTML 5 하이브리드 소스를 사용하였다.

2.1.2 Camera Streaming

카메라 모듈에서 영상을 캡처 할 경우 스트림 데이터의 형태로 저장된다. 이것을 인터넷으로 보내기 위해서는 패킷 형태로 데이터를 전환 할 전용 서버가 필요하다. 이 때 메인 보드 역할을 담당하는 Mbed⁽⁶⁾가 별도의 데이터 전환 기능을 지원하지 않아 따로 이 기능을 지원하는 변환장치를 설치하여야 한다. Figure 2에 카메라 스트리밍을 나타내었다.

본 연구에 사용된 임베디드 컴퓨터인 Raspberry Pi같은 자유도 높은 미니컴퓨터가 이런 역할에 적당하나 리눅스 운영체제를 사용하기 때문에 C언어를 베이스로 사용한다. 그러나 Java Development Kit (JDK)만 설치하면 Java 기반의 라이브 스트리밍 기능과의 호환성도 좋다. 또한 무선 인터넷 USB Dongle을 장착하였기 때문에 별도의 임베디드 용 Wi-Fi 모듈을 거치지 않고 바로 인터넷에 접속할 수 있다. Figure 3은 Raspberry Pi 컴퓨터이다. 스트리밍 쪽에서 Android와 리눅스를 모두 지원하는 Video LAN Client

Figure 2. Camera streaming.

Figure 3. Embedded computer of raspberry pi.

(VLC)라는 미디어 플레이어에 있어 서버 컴퓨터의 호스트 주소만 알고 있다면 컴퓨터나 스마트 폰을 이용해 어디서든 실시간으로 동영상 볼 수 있기 때문에 설치된 장소의 상태를 감시할 수 있다.

2.1.3 Wi-Fi Module

감시로봇에서 Wi-Fi로 데이터를 송·수신하기 위한 것이다. 처음에는 XBee 모듈로 Wi-Fi 접속을 시도했으나 XBee 자체가 원래 Arduino⁽⁷⁾용으로 제작된 모듈이기 때문에 Mbed와 호환을 위하여 인터페이스 기능을 추가하여 송·수신이 가능하게 하였다.

2.1.4 Embedded device

Micro Controller Unit (MCU)는 Mbed NXP LPC1768을 사용하였다. 이 MCU는 많이 사용하는 ATmega⁽⁸⁾와는 달리 웹브라우저를 통한 온라인 컴파일 및 소스 버전 관리 기능, 커뮤니티를 통한 라이브러리 공개와 방대한 개인 위키 페이지, 고성능 32비트 ARM 코어와 그 외 다양한 기능을 제공한다.

Mbed는 따로 설치할 필요가 없는 온라인 컴파일러를 기본으로 제공함은 물론, 개발 중인 프로젝트의 버전관리가 가능하며 온라인 컴파일러이기 때문에 장소 제약 없이 어디서든 작업을 계속할 수 있다.

2.1.5 RPC Remote Control

Asynchronous JavaScript And XML (AJAX) 기술을 활용하여 웹 페이지를 통해 데이터를 받아오는 것뿐만 아니라 로봇을 직접 움직일 수 있으므로 기존 웹 서버상의 요청으로는 할 수 없는 AJAX만의 고유 기능이다.

2.1.6 Sensors

Figure 4는 상온도와 습도를 계측할 수 있는 온·습도 센서 모듈과 회로도이다.

센서 Analog to Digital Converter (ADC)값으로부터 습

Figure 4. Temperature/humidity sensor, circuits (CHIPCAP-D/Module).

도와 온도로 변환하는 방법은 식(1)과 식(2)에 의한다.

$$\%RH = \frac{(V_o - (0.1 \times V_s))}{(0.8 \times V_s)} \times 100 \quad (1)$$

$$T_c = \frac{(V_o - (0.1 \times V_s))}{(0.8 \times V_s)} \times 200 - 50 \quad (2)$$

여기서 V_s 는 입력전압이고, V_o 는 출력전압이다.

Figure 5는 알코올/가스 센서의 회로와 PCB 모듈이다. 대기 중에 있는 알코올 농도는 식(3)으로, 가스량은 식(4)에 따라 데이터를 ARM Processor chip으로 송신한다.

$$C = \frac{A}{P \times r} \quad (3)$$

여기서 C 는 혈중 알코올 농도이고, P 는 음주자의 체중(Kg), r 은 성별계수로써 남자일 경우는 0.7, 여자일 경우는 0.6이다. A 는 섭취한 알코올의 양이며 음주량(ml) × 술의 농도(%) × 0.7894이다.

Figure 5. Circuits of alcohol sensor.

Figure 6. Mbed circuits.

$$V_s(I) = V(I) \times \frac{273}{273 + T_a(^{\circ}\text{C})} \times \frac{(P_a + P_m - P_v)}{760} \quad (4)$$

여기서 V_s 는 표준상태의 가스량이고, P_a 는 대기압, P_m 은 게이지압, P_v 는 포화증기압, T_a 는 가스미터 온도이다.

2.2 하드웨어 회로도

Figure 6은 Mbed 자체의 회로구성이다. Figure 7은 카메라 스트리밍 서버 회로구성으로 웹캠을 사용하여 1인칭 시점이 아닌 3인칭 CCTV형식의 영상을 스트리밍서버를 사용해 앱에서 직접 받아오는 방식을 사용했다.

2.3 동작 Flowchart와 의사코드

Figure 8은 가스 유출 장소에서 감시스마트로봇이 동작하는 Flowchart이며 Table 1은 웹페이지 UI 동작 버튼이다.

버튼 클릭 시 각 버튼 ID를 JAVA Script의 buttonPush 함수로 전달되며, buttonState가 0 (Motor OFF)일 때 buttonState를 1 (Motor ON)로 전환시키고 b값(robot의 움직이는 속도)을 0.25로 전환시킨다. buttonState가 1일 때

Figure 7. Camera streaming server component.

Figure 8. Flowchart.

buttonState를 0으로 전환시키고 b값을 0으로 전환시킨다. button ID가 0, 1, 2, 3에 따라 a값을 “forward”, “left”,

Table 1. Movement Button of Webpage UI

```

IF button Click THEN buttonPush(id)
<script language="javascript">
var buttonState = new Array(4);
var buttonNumber = new Array(1,2,3,4);

FUNCTION buttonPush(id)
{
    IF buttonState[id] is 0
    THEN buttonState[id] = 1; b=0.25;
    ELSE
    buttonState[id] = 0;
    THEN b=0;

    IF id is 0
    a = "forward";
    ELSE IF id is 1
    a = "left";
    ELSE IF id is 2
    a = "right";
    ELSE IF id is 3
    a = "backward";

    var req = new XMLHttpRequest();
    var cmd= "http://" + location.host + "/rpc/m3pi"+"/"
    "+a+", " + b;
    req.open("GET", cmd, true);
    req.send("");
}

```

Table 2. Image View Button of Android UI Main Activity

```

streamPath streamingServerPath;
button1 Click()
{
    Intent ACTION_VIEW;
    Intent streamPath;
}

//MainActivity Sensor and Control Button
//DroidGap in Mbed Activity of Intent.
button2 Click()
{
    Intent Mbed Activity;
}

public class Mbed extends DroidGap
{
    public void onCreate()
    {
        loadUrl("ServerIPAdress");
    }
}

```

“right”, “backward”로 Switching하며 XML Http Request 객체를 생성, AJAX통신기법을 사용하여 각 버튼의 cmd 값을 서버에 송신한다. Table 2는 Android UI Main Activity 영상보기 버튼이다.

String 형으로 스트리밍서버의 Path선언 외부프로그램과 연결하기 위해 ACTION_VIEW를 Intent한다. 그리고 Uri에 미리 선언해둔 streamPath를 지정한 다음 실시간영상을 스트리밍서버에서 받아와 스마트폰내부의 플레이어로 보여준다. Mbed Activity에서 DroidGap를 상속받아 앱 창에 맞게 웹페이지를 Load 한다.

3. 실험 및 고찰

각 파트의 소스들을 코딩할 때 따로따로 코딩하면 잘 실행되었지만 소스들을 하나의 소스로 합쳐서 Compile 할 때는 함수 명칭과 변수 선언이 일치하지 않아 발생하는 Error를 디버깅 하는데 애로사항이 많았다.

처음엔 Html파일을 따로 만들어 Mbed안에 넣어서 웹페이지를 보여주었다. 그러나 각 센서들의 값을 웹페이지에 보여주는데 있어서 클라이언트에서 서버에 요청하는 것이 아니라 서버의 값을 클라이언트에서 가져오는 것이 곤란하였다. 때문에 Html페이지 역시 서버의 코드 안에서 작성해주는 함수를 만들어 센서 값들을 바로바로 띄워 줄 수 있도록 함으로서 문제를 해결하였다.

감시스마트로봇의 움직임을 제어하는 부분은 웹 서버와 비동기적으로 데이터를 교환하고 조작하기 위해 AJAX기술인 Http Request를 사용하여 웹페이지의 각 버튼에 Javascript로 서버에 정해진 값을 전달 해 주도록 하였다. 웹페이지로 제어할 때 앱 없이 브라우저만으로 제어할 수 있는 하이브리드 앱 방식을 사용하였다. 이 방법은 1인칭 시점에서 감시스마트로봇을 제어하지는 못하지만 해상도 면에서 고품질의 영상을 받아 볼 수 있었고 전송속도도 빨라 delay현상을 감소시킬 수 있었다.

라이브 스트리밍에서는 패킷 형태로 데이터를 전환할 전용 서버는 라즈베리파이라는 소형 리눅스 컴퓨터로 하였다. 라즈베리파이의 전압 공급은 Mbed와의 전압 분배를 고려하지 않아도 되는 스마트폰에 사용되는 보조 배터리를 이용하였다.

Android에서는 라이브 스트리밍 영상을 받는 앱을 만들기 쉽지만 센서로 수신하는 데이터와 자동차 모듈의 이동 제어, 영상 스트리밍 정보를 통합하여 관리하는 앱이 필요하여 Android 개발용 컴파일러를 이용해 직접 구성하였다. Http 기반의 스트리밍 소스를 그대로 올리면 Android 기기에서는 데이터가 아닌 인터넷 URL로 인식하여 인터넷 창을 열기 때문에 이를 방지하도록 Real Time Streaming Protocol (RTSP) 방식으로 전환하여 전송하였다.

하드웨어의 최종 외형은 Figure 9와 같다. (a)는 3pi Robot위에 ARM Mbed와 접속을 가능하게 해주는 소켓이

Figure 9. Hardware outside.

Figure 10. Main and web page.

달린 M3pi를 부착시키고 M3pi 확장 키트를 통하여 센서들을 장착한 것이다. (b)는 (a)에 카메라서버를 장착시킨 것으로 라즈베리파이의 전압 공급을 위해 스마트폰의 배터리를 사용하여 필요한 소켓이나 전원부분 회로구성이 이미 갖추어져 있어 별도 회로설계 없이 쉽게 구현이 가능하였다.

Figure 10의 (a)는 Android 어플리케이션 메인 페이지이며 (b)는 지능형로봇을 움직이기 위한 웹 페이지이다. Html형식으로 되어 있어 PC나 모바일을 통해 어디서나 접근이 가능하게 동작시킬 수 있었다. 또한 감시는 스트리밍서버를 통하여 PC나 Android로 웹 카메라영상을 실시간으로 감시 할 수 있다.

4. 결 론

Android 앱을 통해 감시스마트로봇 자체를 조종할 수 있으며 전진, 후진, 좌·우로 이동이 가능하다. 감시스마트로봇에 있는 Mbed에서 수신되는 센서 값은 항상 실시간으로 수신되어 웹서버에 저장되고 웹페이지나 Android 앱으로 실시간 확인되었다. 수신되는 센서 값 중에 특정 상황이 발생하는 경우 감시스마트로봇과 Android 앱에 알림이 전송되어 사용자에게 이를 알려주었고 웹카메라 영상

은 실시간 스트리밍 서버를 통해 언제든지 Android 앱으로 확인할 수 있었다. 또한 고품질영상임에도 빠른 전송속도로 지연현상을 감소시켰으며 전원 분배문제를 해소하였다. Ethernet 통신망을 사용하고 HTTP Server에 AJAX기법으로 감시스마트로봇을 제어하여 인터넷이 연결되어 있는 곳이라면 거리에 제약 없이 어느 곳에서나 감시스마트로봇을 제어하고 데이터를 확인 할 수 있어 감시 및 정찰 기능을 쉽게 수행 할 수 있다. 유 · 무선 공유기 환경은 Mbed내의 서버와 Android 앱의 IP를 설정하고 Port Forwarding을 통해 공유기 내부 망에 접근하면 Android 앱으로도 쉽게 구현할 수 있었다. 그러나 Android로 PhoneGap을 사용하여 웹페이지를 보여주는 방식이 아닌 직접 서버와 통신하는 앱을 만들어 적용한다면 더 정교한 제어가 가능할 것이다.

후 기

본 연구는 2013년 가천대학교의 지원을 받아 수행되었으며 관계제위께 감사드립니다.

References

1. W. S. Um, Y. G. Kim and J. H. Lee, "Development Trend of Intelligent Robots", Journal of Korea Aerospace Industries, Vol. 11, No. 1, pp. 150-160 (2013).
2. S. W. Kim, Y. H. Park and S. G. Kyun, "Implementation of an Intelligent Robot Control System Based on UPnP", Journal of the Korea Institute of Information and Communication Engineering, Vol. 16, No. 10, pp. 2129-2136 and 2234-4772 (2012).
3. I. Y. Chu, "Development Plan of Safety Management on Intelligent Robot", Korea Security Science Association, Vol. 26, pp. 89-119 (2011).
4. C. S. Ki, "Intelligent Robot and Human Face Recognition Convergence Technology", Journal of the Korea Institute of Information Security and Cryptology Vol. 17, No. 5, pp. 25-31 and 1598-3978 (2007).
5. Y. K. Jung and H. G. Cho, "International Standardization Trend of Intelligent Robots", ETRI Journal, Vol. 22, No. 2, pp. 70-78 and 1225-6455 (2007).
6. Ookawa Gaisenuohou, "Transmitting Data for Server Storing up of mbed + Android Meter Communication Program(m)ing Sensor, Smar, Engineering Company, Japan (2012).
7. Mike Riley, "Programming Your Home : Automate with Arduino, Android, and Your Computer", Insite, Seoul Korea (2013).
8. C. S. Chang, "Microprocessor Application and Experience : 80X 86/AVR ATmega128", Chaos Book, Seoul Korea (2013).