

AJAX를 이용한 소방엔진펌프의 모니터링과 제어 시스템 구현

양 오[†] · 이현국

[†]청주대학교 반도체공학과

Implementation of Monitoring and Control System for Fire Engine Pump using the AJAX

Oh Yang[†] and Heon-Guk Lee

[†]Semiconductor Engineering of Cheongju University

ABSTRACT

In this paper, the fire engine pump is controlled and monitored by the AJAX (Asynchronous Javascript and Xml) in the web server. The embedded system with built-in system having a processor and a memory of high performance occurs many problems in transmitting the large amount of data in real time through the web server. The AJAX is different from HTML (Hyper Text Makeup Language) with java script technology and can make RIA (Rich Internet Application). It process the necessary data by using asynchronous and it take advantage of usefulness, accessibility, a fast response time. Using AJAX can build up web server with real time and monitoring that fire engine pump status, check processing pump memory in the event of fire, also remotely monitors can do. The web server system can control the fire engine pump as like the black box. The experimental results show the effectiveness and commercialize possibility.

Key Words : Fire engine pump system, web server monitoring, remote control system, internet of things, AJAX

1. 서 론

현재 소방엔진펌프는 가정이나 산업 어디에서나 많이 사용되고 있다. 그 이유 중 하나는 소방시설 법이라고 할 수 있는데, 소방시설법은 화재와 재난, 재해, 그 밖의 위급한 상황으로부터 국민의 생명, 신체 및 재산을 보호하기 위하여 화재의 예방 및 안전관리에 관한 국가와 지방자치단체의 책무와 소방시설 등의 설치와 유지 및 소방대상물의 안전관리에 관하여 필요한 사항을 정함으로써 공공의 안전과 복리 증진에 이바지하는 것을 목적으로 만들어진 법률이다[1]. 이 법으로 인해서 건축물을 신축, 증축, 이전, 용도변경 등을 할 때 소방시설을 무조건 설치

하도록 되어있다. 그에 따라 불가피하게 소방시설과 함께 물탱크의 물을 공급해줄 수 있는 소방엔진펌프가 자리잡게 되었다.

소방엔진펌프는 건물 화재 시 엔진펌프가 회전하여 물탱크에서 물을 흡수하고, 이를 방수구로 배출하는 역할을 한다. 소방엔진펌프에 의해서 배출된 물은 배관을 통해 건물의 소화전이나 스프링클러까지 흐르게 되고, 화재 발생 시 빠른 초기 진압을 위해 이용된다. 하지만 오래된 건축물이나 제대로 검증되지 않은 소방시설을 갖춘 건물에서 화재가 발생할 때 소방엔진펌프가 작동하지 않을 수 있어 큰 피해를 입을 수도 있다. 이러한 피해를 줄이기 위해 소방엔진펌프를 인터넷에서 실시간으로 모니터링과 제어를 할 수 있도록 한다면, 현재 건물에 관리인이 부재중이더라도 인터넷을 통해 언제나 소방엔진펌프를

[†]E-mail: ohyang@cju.ac.kr

점검하고 제어할 수 있다. 이는 화재 발생 시 소방엔진펌프의 구동 여부를 확인할 수 있고, 화재가 발생된 후 소방엔진펌프가 작동된 시간을 추적하여 화재 원인을 규명할 때도 도움이 될 수 있다. 또한 매번 소방시설 점검을 할 때에도 쉽고 간편하게 펌프의 상태를 확인할 수 있고 쉽게 정기점검을 할 수 있다.

원격 모니터링 및 제어를 위한 내장형 웹 서버 시스템은 네트워크 장비, 네트워크 프린터 등의 하드웨어에 부가 기능으로 원격 고장진단 및 유지보수 등의 목적으로 장치의 컨트롤러에 웹 서버의 기본 기능을 구현한 경우로 일반 웹 서버 시스템에 비해 소형화된 크기와 웹 서버 기능만을 갖고 있으며 원격에서 하드웨어의 상태를 감시하고, 펌웨어 업그레이드, 원격제어 기능을 수행하고 있다[2, 3]. 그 동안 널리 사용되었던 HTML(Hyper Text Markup Language)방식은 웹 서버에 과부하를 가져올 수 있기 때문에 본 논문에서는 AJAX 기술을 이용하여 사용자와 고성능 프로세서로 구현된 웹 서버와 통신하며 빠른 속도로 실시간으로 모니터링과 제어를 할 수 있는 시스템을 구현하였다[4-7].

2. 소방엔진펌프

2.1 소방엔진펌프 시스템

현재 소방산업의 시장은 활발히 성장하고 있고, 소방시설법의 소화 설치 의무화와 안전을 더욱 중시하는 시대의 흐름 속에 소방 산업의 시장 규모는 계속 증가하고 있으며, 옥내 소화전 설비와 스프링클러 설비의 핵심이 되는 소방용 엔진 펌프 제어장치의 신뢰성과 안전성, 편리성이 더욱 중요한 요건이 되고 있다. 소방 엔진 펌프의 고장 유무는 화재 진압에 직접적인 관련이 있고, 생명과도 밀접한 관계를 가지기 때문에 안전점검이 필수이고 손쉽게 안전점검을 할 수 있는 방안이 필요하다. 소방엔진펌프 제어장치의 기본적인 모니터링 시스템 구성도를 Fig. 1에 나타내었다.

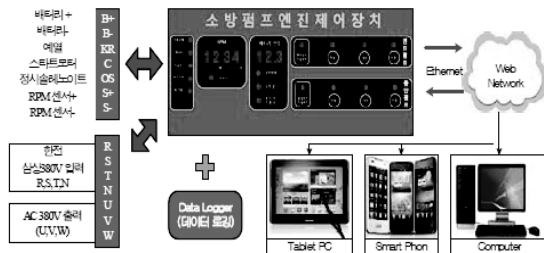


Fig. 1. Remote control and monitoring of fire engine pump.

2.2 소방엔진펌프 모니터링 구성 환경

웹 서버로 모니터링 할 소방엔진펌프는 TEC-ECU1 모델을 사용하였다. TEC-ECU1의 MCU는 ST사의 STM32F429ZE이며 CPU Core는 ARM Cortex-M4F 32bit CPU+FPU이고 2MB Flash memory, 256+4KB SRAM의 메모리와 Ethernet 기능이 내장되어 있다. Fig. 2와 Fig. 3에 각각 웹 서버 기능을 갖는 소방엔진펌프 제어장치의 동작 구성과 제어장치의 Prototype을 나타내었다.

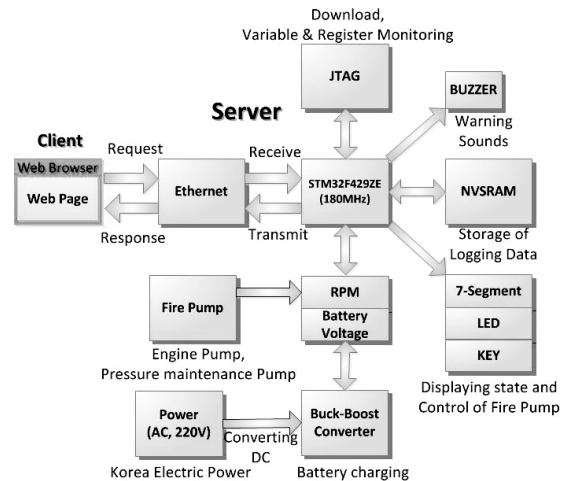


Fig. 2. System configuration of fire pump monitoring.

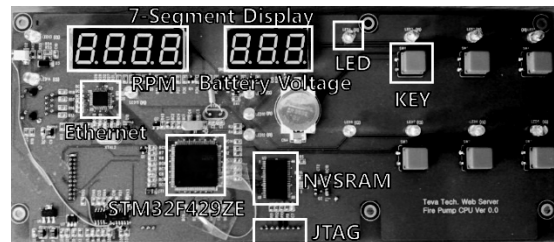


Fig. 3. Prototype of fire pump controller.

3. AJAX

3.1 AJAX 와 기존 방식의 차이

기존의 웹 애플리케이션에서 웹 서버는 클라이언트의 요청을 받아서 검증과 필터링을 거친 후 요청된 내용에 따라서 Fig. 4와 같이 데이터를 가공하여 새로운 웹 페이지를 작성하고 응답하게 된다. 이 때 최초의 양식을 가지고 있던 페이지에 사용자의 양식을 채워 결과물로서 되돌려 받은 페이지는 일반적으로 유사한 내용을 가지고 있는 경우가 많다. 결과적으로 중복되는 HTML 코드를 다시 한번 전송을 받음으로써 많은 대역폭을 낭비하게

된다. 대역폭의 낭비는 응답속도가 느려지는 단점을 가지게 되며 사용자와 대화(상호 반응)하는 서비스를 만들기 어렵게 한다.

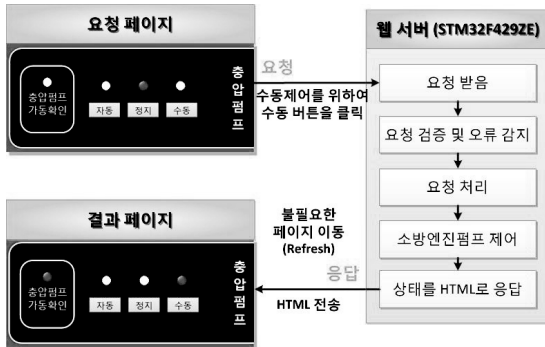


Fig. 4. General HTML process in web server.

반면에 AJAX 애플리케이션은 Fig. 5와 같이 필요한 데이터만을 웹 서버에 요청하여 받은 후 클라이언트에서 데이터에 대한 처리를 할 수 있다. 기존 HTML에서의 대부분 통신 작업은 모두 서버 측에서 했지만 AJAX는 클라이언트 측의 브라우저에서 작업을 수행할 수 있도록 되어있다. 웹 서버에서 전적으로 처리되던 데이터 처리의 일부분이 클라이언트 쪽에서 처리되므로 웹 브라우저와 웹 서버 사이에 교환되는 데이터 량과 웹 서버의 데이터 처리량이 줄어들기 때문에 애플리케이션의 응답성이 좋아진다. 또한 웹 서버의 데이터 처리에 대한 부하를 줄여 주고 뿐만 아니라 요청을 하는 수많은 컴퓨터에 대해서 일어나기 때문에 크게 보면 전체적인 웹 서버 처리량도 줄어들게 된다.

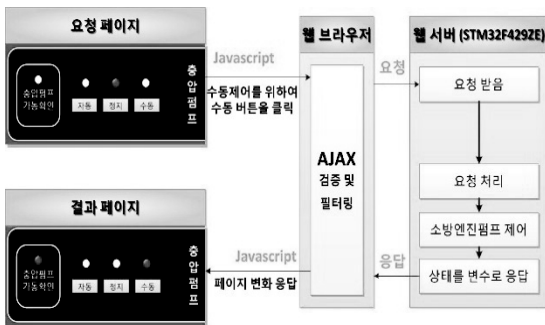


Fig. 5. AJAX process in web server.

3.2 AJAX 장점과 단점

AJAX의 장점은 기존 HTML의 페이지 이동, 페이지 생성과는 다르게 페이지 이동 없이 고속으로 부분화면만

전환이 가능하다는 점과 서버 처리를 기다리지 않고 비동기 요청으로 가능하다는 점이 있다. 그리고 기존 HTML 방식은 클라이언트가 요청을 보내면 서버 측에서 모든 일을 처리한 후 응답하지만 AJAX는 클라이언트의 웹 브라우저에서 먼저 처리를 한 후 서버에 최소한의 통신을 요청 및 응답 받는다. 그러므로 서버 측에서는 처리를 각각의 PC에 분산 가능하고 수신하는 데이터의 양을 줄일 수 있다. 그래서 AJAX 방식을 사용하게 되면 데이터 처리에 대한 부하를 줄일 수 있다는 점을 장점으로 볼 수 있다.

AJAX의 가장 큰 단점으로는 기능을 지원하지 않는 브라우저가 존재하고 HTTP 클라이언트의 기능이 한정된다. 그리고 페이지 이동이 없는 통신으로 인한 보안상의 문제와 지원하는 Charset이 한정되어 있고 스크립트로 작성되므로 디버깅(Debugging)이 용이하지 않다는 점과 요청을 남발하면 역으로 기존의 웹 서버 방식보다 서버 부하가 늘 수도 있다. 그러므로 크로스 브라우저화의 노하우가 필요하고 보안에 더욱 신경을 써야 한다는 점을 단점으로 볼 수 있다.

그러나 정보화시대의 규모가 점점 더 커지고 있고 이용자들이 증가하여 사회가 가속화되고 있기 때문에 서버의 부하를 줄일 수 있고, 소프트웨어 기술로 보안을 강화하여 단점을 보완하고 이용한다면 아주 유용하게 사용할 수 있는 웹 기술이 아닐 수 없다[8].

3.3 실시간 데이터의 AJAX 요청과 응답

클라이언트(Client)측의 웹 브라우저는 소방엔진컴프의 원하는 상태를 페이지 상에 출력하기 위하여 서버에 AJAX요청을 보내게 된다. 그리고 서버는 이 요청을 받아 Client가 원하는 정보를 찾아서 다시 브라우저로 응답을 보낸다. 응답을 받은 브라우저는 Client가 해당 정보를 알아보기 편한 형태로 부분적인 변화를 가져오게 된다.

AJAX 요청을 하기 위해서는 우선 HTTP요청을 할 객체(Object)를 만든다. 여러 브라우저마다 지원하는 기능이 다르기 때문에 XMLHttpRequest를 지원하지 않는 브라우저에서는 ActiveX Object를 이용해야 한다. 그 이후에 만약 HTTP요청이 있을 경우 그 객체는 open 메소드를 통해 GET방식으로 '/ledstate?' 라는 URL(Uniform Resource Locator)에 연결하여 비동기(true)로 요청을 수행한다. 이때 URL은 현재 HTML 페이지를 기준으로 한 상대주소를 의미한다. 이렇게 요청을 보내는 과정에서 객체의 'readyState' 값은 0에서 4로 변화하게 되는데 이때마다 'onreadystatechange' 메소드를 이용하여 객체가 요청완료 상태인지 확인하고(값이 4일 때, 객체는 요청완료 상태) 요청완료 상태일 때 status속성은 200인지(HTTP요청의

GET이나 POST방식은 대개 200을 반환)확인하여 모두 만족하는 요청완료상태가 되면 해당 요청에 대한 URL에서의 응답(처리)을 STM32F429ZE프로세서에서 문자형(char type)으로 반환하게 된다. 이 때 반환된 배열의 문자는 브라우저에서 responseText의 배열로 받아서 그 응답에 대해서 원하는 작업을 수행할 수 있다. 본 논문에서는 반환되는 responseText값에 따라서 웹 페이지의 각 LED 부분이 ON/OFF되도록 구현하였다[9,10].

```
<script language="Javascript">
function LEDstatus()
{
    var LED = false;
    if(window.XMLHttpRequest());
        LED = new XMLHttpRequest();
    else if(window.ActiveXObject)
        LED = new ActiveXObject
            ("Microsoft.XMLHTTP");
    if(LED)
    {
        LED.open("GET", "/ledstate?", true);
        LED.onreadystatechange = LEDcomplete;
        LED.send(null);
    }

    function LEDcomplete()
    {
        if(LED.readyState == 4)
        {
            If(LED.status == 200)
            {
                if(LED.responseText[0] == '1')
                document.getElementById("A1").innerHTML
                = "<img src='control_image/white.png'
                width='15px' height='15px'>";
                else
                document.getElementById("A1").innerHTML
                = "<img src='control_image/red.png'
                width='15px' height='15px'>";
            }
        }
    }
}

```

3.4 웹 서버 모니터링 실험 및 고찰

본 논문의 웹 서버는 AJAX와 HTML을 활용하여 소방엔진펌프의 제어장치를 모니터링 할 수 있도록 STM32F429ZE로 소방엔진펌프의 디스플레이에 표시하는 데이터를 전달해 메모리에 저장을 하고 전달받은 데이터를 0.1초마다 기존의 웹 페이지 전체가 아닌 변화에 해당되는 부분인 배터리 역결선, 엔진 회전속도, 시동실패, 배터리전압의 상태, 엔진·충압펌프의 가동상태 등을 새로 고침 하게 된다. 그리고 NVSRAM메모리에 저장되어 있는 데이터를 가지고 웹 서버의 동작내역을 사용자가 원하는 기간을 선택하여 확인할 수 있다. 또한 일반 클라이언트들은 자신의 소방엔진펌프의 동작내역과 실시간 모니터링을 할 수 있지만 관리자에게만 아이디와 비밀번호를 부여하여 펌프를 제어할 수 있도록 함으로써 보안성을 높였다.

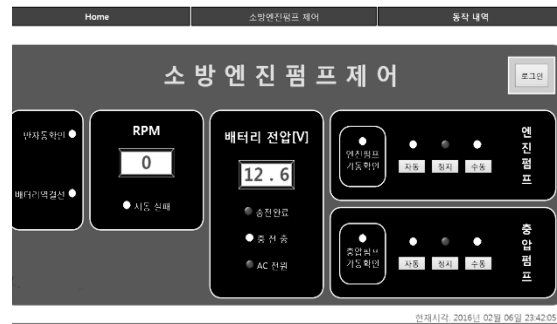


Fig. 6. Web page for monitoring and control.

Fig. 6은 실시간 소방엔진펌프의 상태를 모니터링 할 수 있는 웹 페이지로써, 내부 메모리에 저장된 ID와 Password를 알고 있는 관리자만이 소방엔진펌프를 제어할 수 있도록 구축된 웹 페이지이다. 로그인한 관리자는 ID와 Password, RTC(Real Time Clock)를 변경할 수 있다. 그리고 엔진·충압펌프의 자동, 정지, 수동을 마음대로 제어할 수 있다.

아울러 Fig. 7은 소방엔진펌프에 대한 데이터 로깅 시스템으로 소방엔진펌프의 일정한 기간 동안 동작된 결과를 확인할 수 있는 웹 페이지이다. 이러한 데이터는 NVSRAM에 15년간 저장되도록 하여 장기간 사용이 가능하다. 데이터 로깅은 소방엔진펌프의 작동시간을 추적하여 화재의 원인규명에 도움이 되고 일정 시간마다 점검하는데 편리한 기능을 제시할 뿐만 아니라 유지보수에 따른 경제성을 향상 시킬 수 있을 것으로 예상된다.

Home		소방엔진펌프 제어		동작 내역				
조회기간 2016년 01월 01일 ~ 2016년 12월 31일 조회								
날짜	시간	연진펌프	송압펌프	배터리결선	시동상태	AC전원	RPM	배터리전압
2016-01-03	06 : 10 : 26	정지	정지	정상	정상	OFF	0	0.5 V
2016-02-12	06 : 16 : 06	정지	정지	정상	정상	OFF	0	0.0 V
2016-03-15	06 : 13 : 18	정지	정지	정상	정상	OFF	0	0.5 V
2016-04-07	06 : 21 : 25	정지	정지	정상	정상	ON	0	14.1 V
2016-05-10	06 : 46 : 35	정지	정지	정상	정상	ON	0	12.9 V
2016-06-01	06 : 37 : 26	정지	정지	정상	정상	ON	0	14.1 V
2016-07-20	06 : 05 : 01	정지	정지	정상	정상	ON	0	12.6 V
2016-08-09	07 : 16 : 20	정지	정지	정상	정상	ON	0	14.2 V
2016-09-03	07 : 27 : 04	정지	정지	정상	정상	ON	0	12.9 V
2016-10-14	06 : 51 : 36	정지	정지	정상	정상	ON	0	14.2 V
2016-11-08	06 : 38 : 19	정지	정지	정상	정상	ON	0	12.6 V
2016-12-12	06 : 47 : 23	정지	정지	정상	정상	ON	0	14.2 V

Fig. 7. Experimental result of data logging system.

기존의 HTML방식과 AJAX 방식의 속도차이를 확인하기 위하여 측정된 통신속도 파형은 Fig. 8과 Fig. 9에 나타내었고, 이를 분석하여 정리한 실험 결과는 Table 1과 Table 2와 같다.

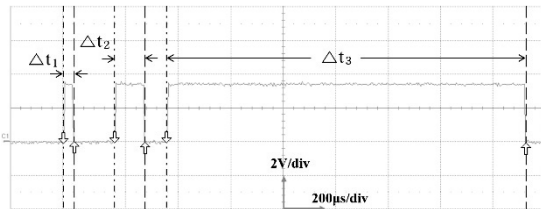


Fig. 8. Result for communication speed with refresh.

Table 1. Experimental results for HTML

구간	설명	시간
Δt_1	HTML방식으로 웹 페이지에서의 요청을 검사하는 시간	42 μ s
Δt_2	HTML방식으로 소방엔진펌프의 전면 디스플레이 데이터를 불러오면서 제어 페이지에 접속하는 시간	124 μ s
Δt_3	HTML방식으로 데이터 조회와 함께 동작내역 페이지에 접속하는 시간	1265 μ s

Table 1에서 보이는 것과 같이 웹 서버에서 HTML방식으로 웹 페이지의 요청을 받았을 때, 그 요청을 검사하는 시간 Δt_1 은 42 μ s로 측정되었다. 그리고 소방엔진펌프 제어 페이지와 전면 LED 및 7-Segment의 정보를 모두 가져오는 시간 Δt_2 는 124 μ s이고, 동작내역 페이지 응답과 데이터 로깅을 동시에 수행할 때 걸리는 시간 Δt_3 는 1265 μ s만큼 소요 됐다.

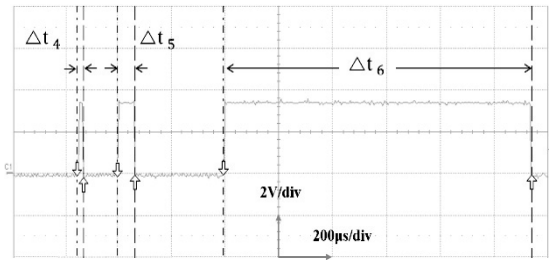


Fig. 9. Communication Speed waveform with AJAX.

Table 2. Experimental results for AJAX

구간	설명	시간
Δt_4	AJAX방식으로 웹 페이지에서의 요청을 검사하는 시간	26 μ s
Δt_5	AJAX방식으로 제어장치 전면 디스플레이의 모든 데이터를 소방엔진펌프 제어 페이지로 불러오는 시간	66 μ s
Δt_6	AJAX방식으로 동작내역의 데이터를 조회하고, 데이터를 NVSRAM에서 찾아 페이지로 불러오는데 걸리는 시간	1144 μ s

Table 2에서는 AJAX를 이용하여 페이지의 부분적인 변화만 주었을 경우의 실험 결과를 나타내었다. Δt_4 에서는 HTML방식으로 웹 페이지의 요청을 검사했을 때보다 16 μ s 더 빠르게 동작하였다. AJAX로 제어장치의 전면 디스플레이의 데이터 부분만 불러올 경우 HTML방식보다 58 μ s의 시간이 절약되었고, 동작내역 페이지의 데이터를 부분만 변화시켜 데이터를 로깅한 경우의 시간은 121 μ s 감소하였다. 이와 같이 AJAX방식을 이용하면 더욱 빠르게 통신할 수 있음을 알 수 있다.

4. 결론

본 논문에서는 소방엔진펌프의 원격 모니터링 및 제어를 위해 HTML방식보다 빠른 AJAX기술을 이용하여 실시간으로 각각의 상태 부분만 변화를 보이도록 하였고, 관리자만이 알 수 있는 정해진 ID와 비밀번호로 로그인 하면 실시간 제어가 가능하도록 하여 외부 클라이언트가 임의로 제어하는 것으로부터 보호되도록 구현하였다. 그리고 소방엔진펌프에 특별한 이상이 생길 때마다 그 때의 주요 상태들을 기록해놓도록 구성하여 동작내역 페이지에서 이를 확인할 수 있도록 구현하였다. 인터넷으로 언제 어디서나 소방엔진펌프를 실시간으로 모니터링하고 관리자는 제어까지도 할 수 있다. 이는 관리자가 건

물에서 자리를 비운 사이 화재가 일어났을 때 화재의 원인 규명, 소방엔진펌프의 오동작 원인 등을 파악하는데 편리할 것이다.

참고문헌

1. Nation security department, "Fire prevention, Law about safety management and having fire protection system", The Legislative Office, p. 1, January 2016.
2. Kyongho Han, "Implementation of Embedded Micro Web Server for Web based Remote Hardware Control and Monitor", Journal of the Korean of Illuminating and Electrical Installation Engineers, Vol. 20, No. 6, pp.104~110, July 2006.
3. Seongrak Rim, "An Embedded Linux Web Server for Remote Monitoring & Controlling the Internet Appliance", Journal of Information Technology Applications & Management, Vol. 12, No. 3, pp.124-131, December 2005.
4. L.D Paulson, "Building rich web applications with Ajax", IEEE Computer Society, Vol. 38, No. 10, pp.14-17, Oct 2005.
5. S. Salva, Univ. LIMOS, Aubiere.d'Auvergne, and P. Laurencot, "Automatic Ajax Application Testing", Internet and Web Applications and Services, 2009.
6. Junyong Woo and Eunju Kim, "Application of Web using the Ajax and PHP", Korea Information Science Society from Kangwon-do, pp.673-676, 2008.
7. Seonghwan Kim, Janghyeon Park, and Cheolhun Na, "Embedded System with remote monitoring using the Ajax interface web-server", Science announcement Papers of Korea Intelligence System Society, Vol. 21, No. 2, pp.185-186, December 2011.
8. Woonho Park, Heonguk Lee, and Oh Yang, "Implementation of Fire Pump Monitoring and Control System using the Web Server", Information and Control symposium ICS, pp.179-181, April 2016.
9. Hyosang Kweon and Oh Yang, "Realization of monitoring system with solar inverter using the Ajax", Korea Information and Communication Society Paper, Vol. 16, No. 9, pp.1915-1922, September 2012.
10. Morrison Michael, "Ajax Construction Kit: Building Plug-and-Play Ajax Applications", Acorn Web professional, pp.77-100, March 2011.

접수일: 2016년 8월 29일, 심사일: 2016년 9월 8일,
 게재확정일: 2016년 9월 9일