

주방 가스레인지 빌트인 방식에서 가스 누출검사를 위한 IoT 검사 시스템 개발에 관한 연구

강대국, 최영규*

A Study on the Development of IoT Inspection System for Gas Leakage Inspection in Kitchen Gas Range Built-in Method

Dae Guk Kang, *Young Gyu Choi

요약 본 연구에서는 가정 내의 가스 밸브에 장착되는 가스 타이머와 ESP-01 와이파이 모듈을 활용하여 서버와 연동되는 IoT 검사 시스템을 개발하였다. 가스 누출 IoT 검사 시스템의 서버 환경은 APM(Apache, PHP, MySQL)을 설치하여 세대별 가스 압력 데이터를 수집하여 누출 체크를 할 수 있도록 하였다. 가스 누출 IoT 검사 시스템 제어를 위해 앱 인벤터를 활용하여 가스 누출 체크 값을 실시간으로 관리를 할 수 있도록 하였다. 또한 모바일 앱으로 회원 관리 및 WiFi 설정, 누출 체크 값 등을 확인할 수 있도록 사용자의 편의성을 높였다. 지역별 가입자 관리를 하기 위해서 관리자 모드로 로그인하여 사용자 목록 확인을 하여 누출 검사 여부와 결과에 대한 안내를 할 수 있도록 하였다. 또한 사용자가 가스누출 체크 버튼을 누르면 자동으로 압력을 체크하여 서버에 측정값이 저장되고 가스 누출 발생 시에는 경보 및 수리 후 누출 체크를 하여 정상이면 사용할 수 있도록 하였다. 또한 회원의 중복 가입을 막기 위해 MAC 주소를 기반으로 회원 관리를 할 수 있도록 하였다.

Abstract In this study, an IoT inspection system that can be linked with a server was developed using a gas timer and ESP-01 Wi-Fi module installed on a gas valve in the home. The server environment of the gas leak IoT inspection system was installed with APM (Apache, PHP, MySQL) to collect gas pressure data by generation so that leakage checks could be performed. In order to control the gas leak IoT inspection system, the app inventory was used to manage the gas leak check value in real time. In addition, user convenience has been enhanced so that membership management, WiFi settings, and leakage check values can be checked through mobile apps. In order to manage subscribers by region, the user list was checked by logging in in the administrator mode so that the information on whether or not the leak test was conducted and the results could be provided. In addition, when the user presses the gas leak check button, the pressure is automatically checked, and the measured value is stored in the server, and when a gas leak occurs, the leakage check is performed after alarm and repair so that it can be used if normal. In addition, in order to prevent overlapping membership, membership management can be performed based on MAC addresses.

Key Words : Apache, ESP-01, IoT, MySQL, MAC address, PHP

1. 서론

최근 전자 제어가 가능한 제품군 대부분에서 사

물(Thing)이 통신을 하는 IoT 정보통신기술 을 도입함으로써 인간의 개입을 최소화 하면서 사물간에 정보를 수집, 처리, 교류 소통하는 시대로 다양

This was supported by Korea National University of Transportation in 2021

*Corresponding Author : Department of Computer Engineering, Korea National University of Transportation(ygchoi@ut.ac.kr)

Received August 08, 2022

Revised August 17, 2022

Accepted August 26, 2022

한 인터넷 통신 프로토콜을 통해 지능형 네트워크로 발전하고 있다.

본 연구에서는 기존 가스 누출 검사를 검침원이 세대를 방문하여 누출 체크를 하는 방식에서 스마트 기술로 세대별 가스 압력 데이터의 수집 분석을 통해 가스 누출 사고를 미연에 방지하 방지하기 위한 IoT 검사 시스템에 대한 연구를 진행하였다.

2. 시스템 설계

2.1 시스템 구성요소

시스템의 구성 요소는 인터페이스, 네트워크, 장치로 그림 1과 같은 구조로 이루어져 있으며 각 요소가 유기적으로 연결되어 사용자의 ID, 등록된 장치의 식별 번호, 가스 누출 검사 등의 정보를 공유할 수 있도록 하였다.

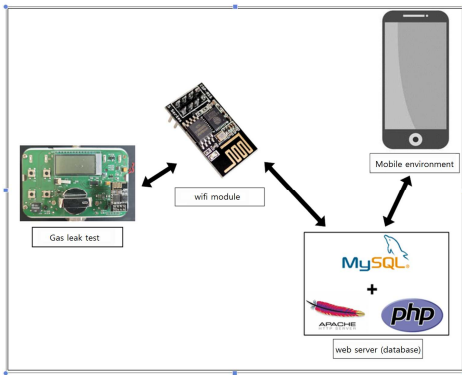


그림 1. 제안된 시스템 구조
Fig. 1. Proposed system architecture

가스누출 검사를 위한 IoT를 적용한 검사 시스템에서 압력 센서 측정 값은 UART 통신을 통해 와이파이 모듈이 수신하며 데이터는 와이파이 모듈(Client)의 Web Server 접속을 통해 MySQL의 쿼리문으로 데이터 삽입, 갱신을 그림 2와 같은 과정으로 동작하도록 하였다.

본 연구의 시스템에서는 장치를 이용하여 가스 누출 검사를 실시하면 센서가 측정하는 정보를 수신하여 누출 여부를 검출하고 앱 매체를 통해 누출에 대한 정보를 직관적으로 확인이 가능 하도록 하

였다.[1]

모바일 앱이 데이터베이스의 정보를 수신할 경우 와이파이 모듈과는 별개로 동작하며 와이파이 모듈이 서버에 삽입하거나 갱신하는 데이터베이스의 테이블 내부의 최종 데이터 값을 실시간 조회가 가능하도록 하였다.[2][3]

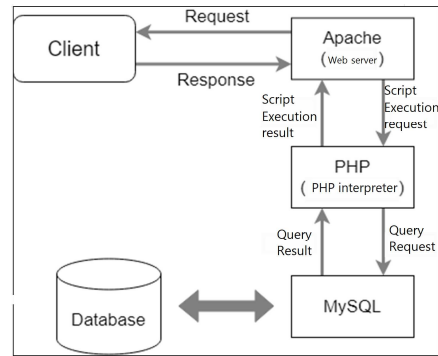


그림 2. 웹서버 구축 및 동작
Fig. 2. Web server flow chart

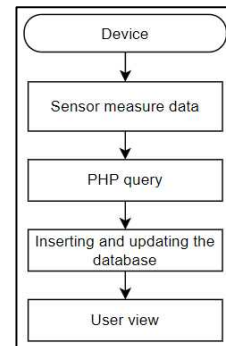


그림 3. 데이터 처리 플로우 차트
Fig. 3. Web server flow chart

그림 3은 누출 검사를 위한 센서 데이터가 MySQL과 연동되는 PHP의 쿼리문을 통해 데이터베이스에 삽입 및 갱신되고 사용자에게 제공하는 흐름을 보여준다. PHP 코드의 구현은 모바일 앱의 각 동작 버튼마다 정해진 쿼리문을 수행하여 문제 발생 시 수정이 용이하게 하였으며 제안된 시스템은 가정의 가스 배관 내의 압력 측정을 통해 데이터 수집 및 저장하여 가스 안전관리를 위한 모니터

링 시스템으로서 활용할 수 있도록 하였다.

2.2 데이터베이스 구성

가스 누출 IoT 검사 시스템의 세대별 데이터와 관리자의 안전 관리를 위한 서버 데이터베이스 구성은 표 1과 같다.

표 1. 데이터베이스 구성
Table 1. Database Configuration

테이블 명	포함 데이터
login	Email, name, password, MAC, residential address
logs	Email, MAC, calculated sensor information, time
userlogs	MAC, Measurement sensor information, measurement time

세대별 회원 가입을 하는 경우 'login' 테이블에 이메일, 이름, 비밀번호가 등록되고 누출 체크 가스타이머 등록 시에는 와이파이 모듈과 연결된 와이파이의 MAC 주소와 거주 주소가 등록되며 MAC 주소는 세대를 분별하기 위해 각 테이블에 존재한다. 또한 가스 누출 검사를 수행할 경우 측정 센서 정보가 'userlogs' 테이블에 저장되며 세대를 식별하여 평균으로 계산된 정보가 'logs' 테이블에 갱신하도록 하였다.

3. 시스템 구현

3.1 모바일 앱 구현

모바일 웹의 초기화면은 로그인 화면과 회원 가입 화면 구성은 그림 4와 같이 세대별로 회원 가입을 위해 '이름', '이메일', '비밀번호'를 입력한 후에 등록하도록 앱 인터페이스로 구현하였다. 초기 설정에서는 기기 등록 과정을 거치지 않아 데이터베이스의 'login' 테이블에 MAC 주소와 세대별 주소가 존재하지 않는다.

회원 가입 후 'login' 테이블에는 '이름', '이메일', '비밀번호' 정보만 존재하며 이 상태로 로그인을 진행하면 'login' 테이블에 MAC 주소가 등록되지 않았기 때문에 기기 등록 과정을 진행하여 세

대를 구분할 수 있도록 하였다.

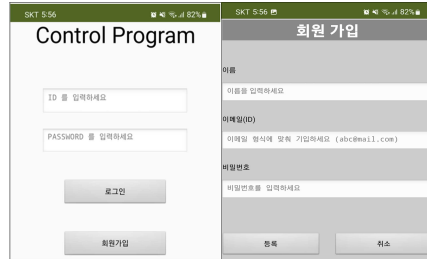


그림 4. 메인 로그인과 회원가입 화면
Fig. 4. main login and membership screen

그림 5은 기기를 등록하는 과정에서 주소 API (원격정보서비스)를 활용하여 거주지 주소를 검색하여 입력 가능하도록 하였다. 주소 설정 후에는 'login' 테이블에 주소 정보를 삽입하는 과정을 거친다.

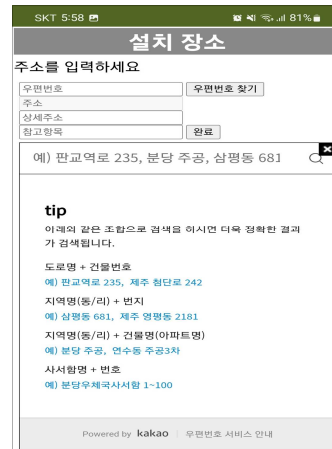


그림 5. 모바일 앱 주소등록 화면
Fig. 5. Mobile app address registration screen

그림 6은 와이파이를 조회 및 선택하고 연결하는 과정을 보여주며 세대별 설치된 가스누출 검사 장치의 ESP-01 WiFi 모듈과 서버가 연동할 수 있도록 연결하며 모듈과 연결된 와이파이의 MAC 주소 정보를 'login' 테이블에 등록하도록 하였다.



그림 6. WiFi 연결 및 mac주소 등록 화면
 Fig. 6. WiFi connectivity and mac address registration screen.

그림 7은 세대별 가스 누출 검사를 진행한 후 검사 결과 조회의 결과 값을 나타내었다.[4]



그림 7. 사용자 정보 조회 화면
 (누출검사 실시 그래프)
 Fig. 7. User Information Inquiry Screen
 (leak test execution graph)

그림 8은 세대별 회원가입, 로그인, 기기등록, 데이터 조회 등 과정을 순서도로 나타내었다.

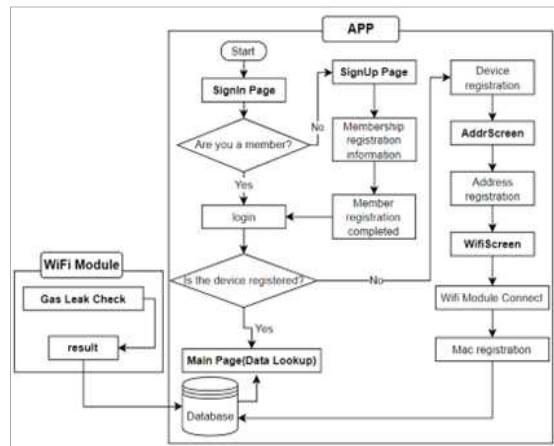


그림 8. 사용자 모바일 앱 이용 순서도
 Fig. 8. User's mobile app usage

그림 9는 가스 공급 업체 서버 관리자가 세대별 누출 검사 시간 및 누출 검사 결과를 출력하여 확인할 수 있도록 하였다.

그림 10은 관리자 페이지에서 가스 누출 검사를 진행한 세대와 진행하지 않은 세대를 구분하여 확인 및 가스 누출 검사를 진행하도록 하였다.

번호	계정	누출 감지	측정 시간
1	example	0	2022-06-19 05:25:43
2	example2	0	2022-06-19 05:52:03

그림 9. 관리자 페이지 사용자 목록
 Fig. 9. Administrator Page Users List

3.2 와이파이 모듈 구현

와이파이 모듈과 연결되어 있는 장치가 모바일 앱을 통해 설치 장소 내 와이파이의 정보를 입력받아 네트워크에 연결이 되면 와이파이 모듈은 앱과

별개로 서버와 통신하며 PHP의 쿼리문을 수행하며 데이터를 삽입 및 갱신한다.

계정	이름	주소	누출 검사
example2	example2	충북 충주시 대소원면 대학교 50 한국교통대학교	미 실시
example	example	충북 충주시 대소원면 대학교 50 한국교통대학교	미 실시

그림 10. 관리자 페이지 가스 누출 검사 관련 목록
Fig. 10. Administrator Page List of Gas Leak Inspections

그림 11과 같이 누출 검사 측정 데이터의 처리 과정으로 장치에서 측정되는 센서 데이터는 UART 통신을 통해 와이파이 모듈로 전송되고 데이터베이스에 정상적으로 삽입이 된다.

그림 12는 가스 누출 검사 밸브 장치에서 송신하는 문자열 데이터 센서 값에서 유효한 값이 있는 경우 값이 들어오는 것으로 판단하고 변수에 저장되는 문자열 데이터 형식은 “ADC4 = (센서 값)\n”로 앞의 문자열을 기준으로 센서 값을 저장한다.

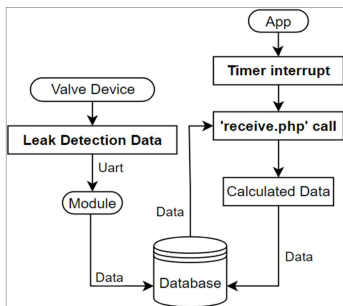


그림 11. 누출검사 데이터 처리
Fig. 11. Leak test data processing

또한 그림 13은 실시간으로 변수에 저장되는 데이터는 MAC 주소와 함께 서버의 센서 데이터 삽입 PHP를 호출하여 ‘userlogs’ 테이블에 삽입하며 측정시간 필드에는 삽입 순간의 시간을 기록하여 측정 시간의 확인이 가능하도록 하였다.

```

void read_sensordata(){
    if(Serial.available()>0){
        uartreadBuffer[uartreadcheck] = Serial.read();
        uartreadcheck++;
        if(uartreadBuffer[uartreadcheck-1]!='\n'){
            sscanf(uartreadBuffer,"%s = %s", mode, data);
            if(0==strcmp(mode,"ADC4")){
                String adcset(data);
                nowAdc = adcset;
            }
            uartreadcheck = 0;
            uartreadBuffer[0]= '\0';
            mode[0] = '\0';
            if(nowAdc!="0"){
                init_adcData(apmac, nowAdc);
            }
        }
    }
}
    
```

그림 12. 센서 값 수신 코드
Fig. 12. Sensor value reception code

앱에서는 주기적으로 동작하는 타이머 인터럽트로 인해 센서 값의 평균치에 대한 차 값을 계산하는 PHP 코드를 계속 호출하여 ‘userlogs’ 테이블의 측정 센서 값을 ‘logs’ 테이블 사용자의 MAC 주소와 매치시키고 계산된 데이터 값을 갱신하도록 하였다.

```

<?php
include("../preset.php");
$mac = $_GET['mac'] ?? '';
$sensor = $_GET['sensor'] ?? '';
$ret = mysqli_query($connect,"INSERT INTO
userlogs(mac,sensor) VALUES ('$mac','$sensor')");
if($ret){}
?>
    
```

그림 13. 센서 값 데이터베이스 삽입 코드
Fig. 13. Sensor Value Database Insertion Code

3.3 서버 부하 테스트

서버의 부하를 테스트하기 위해 일반적으로 웹 서버의 성능 테스트를 위해 하는 접속을 강제로 늘려서 부하를 주는 스트레스 테스트를 진행하였다. 또한 수행은 AB라고 하는 Apache Bench를 이용하여 성능 검사를 위해 보내는 요청 수와 동시에 접속하는 수를 조정하며 수행하였다.

```

<?php
    $aret = mysqli_query($connect,"SELECT * FROM
    userlogs WHERE mac='$mac' ORDER BY
    idx DESC LIMIT 20 OFFSET 20");
    if(mysqli_num_rows($aret)=20){
        while($rows = mysqli_fetch_array($aret)){
            $sensors[] = $rows["sensor"];
        }
        if(array_key_exists("19",$sensors)==FALSE){ }
        else{
            $asum = 0;
            for($i=0; $i<=19; $i++){
                $asum+=$sensors[$i];
            }
            $aavg = $asum/20;
            $a = round($aavg);
        }
    }
    $bret = mysqli_query($connect,"SELECT * FROM
    userlogs WHERE mac='$mac' ORDER BY
    idx DESC LIMIT 20 OFFSET 0");
    if(mysqli_num_rows($bret)=20){
        while($rowss = mysqli_fetch_array($bret)){
            $sensorsb[] = $rowss["sensor"];
        }
        if(array_key_exists("19",$sensorsb)==FALSE){ }
        else{
            $bsum = 0;
            for($i=0; $i<=19; $i++){
                $bsum+=$sensorsb[$i];
            }
            $bavg = $bsum/20;
            $b = round($bavg);
        }
    }
    echo "\n";
    mysqli_query($connect,"UPDATE logs SET sensor
    = '$sensorret' WHERE mac = '$mac'");
?>

```

그림 14. 누출 체크 알고리즘
 Fig. 14. Leak Check Algorithm

본 테스트의 시나리오는 전체 요청 수를 처리하는데 동시 요청의 변화에 따라 얼마만큼의 성능 변화가 있었는지 모니터링하였다. 페이지의 요청은 웹서버를 통해 PHP의 쿼리문 실행으로 확인하였다.

그림 15는 동시 요청에 따라 변화하는 응답시간의 서버 부하 테스트 결과를 그래프로 나타내었다. 공급업체 서버시스템은 세대별 누출 검사 장치로부터 데이터를 수집하여 결과 값을 기반으로 가스 누출 상태를 조회하기 위한 시스템으로 누출 검사

장치 동작에 영향을 주지 않는다.

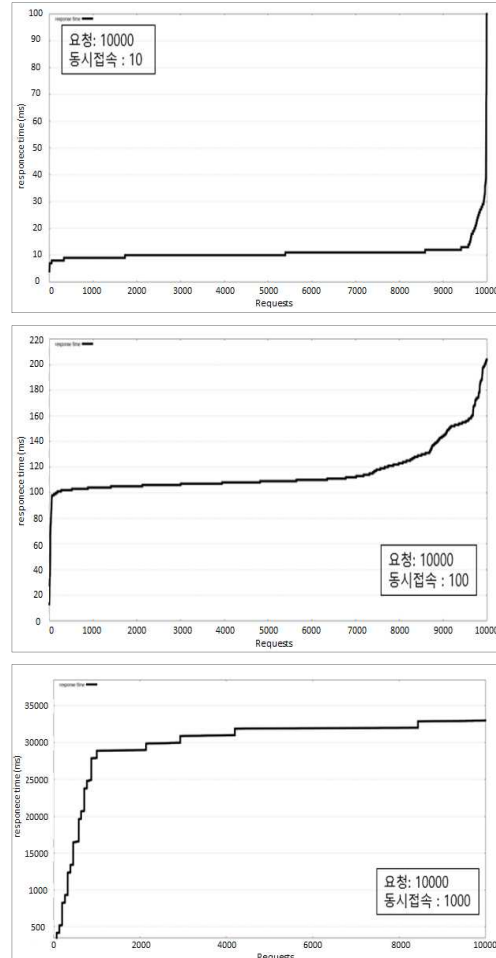


그림 15. 서버 부하 테스트 그래프(요청과 응답시간)
 Fig. 15. Server Load Test Graph
 (Request and Response Time)

따라서 표 2에서 부하 테스트 서버 모니터링에서 300~10,000 세대가 동시 접속을 해도 개별 처리 속도는 약 30ms 정도에서 동작하므로 아파트 단지와 같이 구획을 지정하여 시스템이 구축될 경우 세대 전체가 동시에 접속하는 것은 아니기 때문에 가스 누출 검사를 위한 IoT 검사 시스템의 동작에 문제가 없음을 확인하였다.

표 2. 서버 부하 테스트 결과

Table 2. Server load test result

Request	simultaneous connection	Total Time(sec)	Fail	Processes per second	Individual processing speed(ms)
100	1	0.549	0	182.04	5.493
1000	1	5.016	0	199.37	5.016
10000	1	49.220	0	203.17	4.922
10000	10	13.169	0	759.35	1.317
10000	100	11.680	0	856.20	1.168
10000	300	308.612	0	32.40	30.861
10000	500	309.689	0	32.29	30.969
10000	1000	308.652	0	32.40	30.865
10000	10000	310.566	0	32.20	31.057

4. 결론

일반적으로 도시가스 누출 점검은 검침원을 통해 일정 주기로 세대를 방문하여 검침이 이루어지기 때문에 방문시간을 맞추기가 어렵고 사용자가 검사할 수 있는 방법이 없기 때문에 가스 누출 발생 시 가스안전에 취약할 수 있다.

본 연구에서는 사물인터넷을 이용하여 가스 누출 검사 및 가스안전관리 서버를 구축하고 IoT 가스 누출 검사 시스템의 구현을 위해 서버 구축은 PHP 호출 코드를 분리하여 유지보수 및 개발 소요를 적게 프로그램을 개발 하였으며 ESP-01 직렬 와이파이 무선 모듈 장치를 누출 검사 장비에 내장할 수 있도록 개발된 누출 검사 장비로 동작 시험한 결과 10,000 세대가 동시에 접속을 해도 개별 처리속도가 31ms로 가스 누출 검사 및 세대별 관리는 하는데 사물인터넷(IoT) 기반의 검사 시스템이 안정적으로 동작함을 검증하였다.

향후 연구과제는 가스공급업체 및 세대별 기능적 요구사항 및 보안등에 관한 연구를 진행할 예정이다.

REFERENCES

[1]*Young Gyu Choi "A Study on the Development of Low Power Automatic ON/OFF Valve System for Gas Leak

Detection", The Korea Institute of Information, Electronics, and Communication

[2]Dae Guk Kang, Ji hun Kwak Young Gyu Choi "A study on valve actuator control using Wi-Fi module", The Korea Institute of Information, Electronics, and Communication Technology Summer Comprehensive Academic Presentation symposium Vol.13 No.1, pp.57-60, 2020

[3] Hae Keon Lee, Dong Kurl Kwak, Bong Seob Lee "A Study on Real-time Monitoring System of Indoor Fire Detector using IoT", The Korean Institute of Power Electronics, a collection of academic papers 2018-07, pp267-269, 2018

[4] Kuo-Pao Yang, Patrick McDowell, Pratyush Devkota, Silvin Pradhan, Roshan Bhandari, Zachary Madewell "Detecting Gas Leaks: A Case Study in IoT Technologies", European Journal of Engineering and Technology Research Vol 6 No 7, pp103-106, 2021

[5]Fadhilah Aman, Thunesver Puvanar Thiran, Khairul Huda Yusof, Norazliani Md Sapari "IoT Gas Leakage Detection, Alert, and Gas Concentration Reduction System", IEEE 12th Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp55-60, 2022

저자약력

강 대 국 (Dae Guk Kang)



- 2022.08 현재 한국교통대학교 컴퓨터공학과 석사 수료

<관심분야>

Embedded System & ASIC Design

* 최 영 규 (Young Gyu Choi)



- 1992.5~2022.8 현재 한국교통대학교 컴퓨터공학과 교수

〈관심분야〉

Embedded System & ASIC Design