

아두이노와 파이썬 웹서버를 이용한 스마트홈 오토메이션*

고지혜 김예찬⁰ 이향원

건국대학교 인터넷미디어공학과

minimind@naver.com, yeachan10@konkuk.ac.kr, leehw@konkuk.ac.kr

Smart Home Automation using Arduino and Python Web Server

JiHae Ko YeaChan Kim⁰ Hyang-Won Lee

Internet & Multimedia Engineering, Konkuk University

요약

IoT의 시대가 도래하면서 스마트 홈에 관한 관심이 높아지고 있고, 이와 관련된 주제의 연구가 활발히 진행되고 있다. 특히 아두이노와 같은 마이크로 컨트롤러를 이용한 프로젝트가 많이 존재하는데, 이러한 프로젝트들의 통신 및 성능 쪽의 문제점들을 개선하기 위해 아두이노와 사용자 사이에 웹 서버를 두어 통신하는 스마트홈 시스템을 개발하였다. 본 논문에서는 해당 시스템을 소개한다. 웹 서버를 이용함으로써 사용자가 원거리에서 안드로이드 기기를 이용하여 집안의 에어컨, 조명등의 사물들을 제어할 수 있고, 더 나아가 행동 패턴을 분석하는 서비스도 개발하였다.

1. 서론

IoT 서비스, 특히 스마트 홈에 대한 사람들의 관심은 해가 갈수록 높아지고 있고, 이와 관련된 주제의 연구가 활발히 진행되고 있다. 그리고 그에 따른 시장 규모 역시 2020년에는 600억 달러가 될 것이라 예상되고 있다.

또한 이와 관련된 아두이노와 같은 오픈소스 하드웨어 및 소프트웨어를 기반으로한 프로젝트가 많이 있다. 그리고 아두이노로 구축한 스마트 홈 시스템들은 아두이노나 기타 마이크로 컨트롤러를 서버로 하여 사용자와 통신하게끔 구현한 경우가 대부분이다.

이 경우 아두이노의 용량, 성능 등의 문제로 데이터 처리가 많은 작업들을 원활하게 할 수 없다. 다른 통신 같은 경우에도 같은 AP(Access Point)에서 WiFi나 블루투스 등 근거리 통신방식을 지원하지만 원거리 통신이 불가능하다.

이러한 문제를 해결하기 위해서 사용자와 아두이노 사이에 웹서버를 두어 사용자가 원거리에서 스마트폰을 이용하여 원거리 통신을 가능하도록 하였다. 아두이노 사이의 근거리 통신 같은 경우엔 IEEE 802.15.4의 근거리 저전력 무선표준 프로토콜을 기반으로 한 Xbee 통신을 이용하였다.

이 논문에서는 이러한 통신 방법을 이용해서 사물을 제어하는 서비스를 제공하는 것을 목표로 한다. 또한 웹서버를 이용하여 데이터를 저장하고, 처리할 수 있으므로 사용자의 제어 기록과 센서값(온도, 위치)을 기반으로 한 클러스터링 알고리즘과 Gaussian Distribution 모델

링을 통하여 사용자의 패턴을 분석하고 그 패턴에 따른 자동제어서비스를 제공하는 것을 목표로 한다

2. 시스템 구성 및 구현

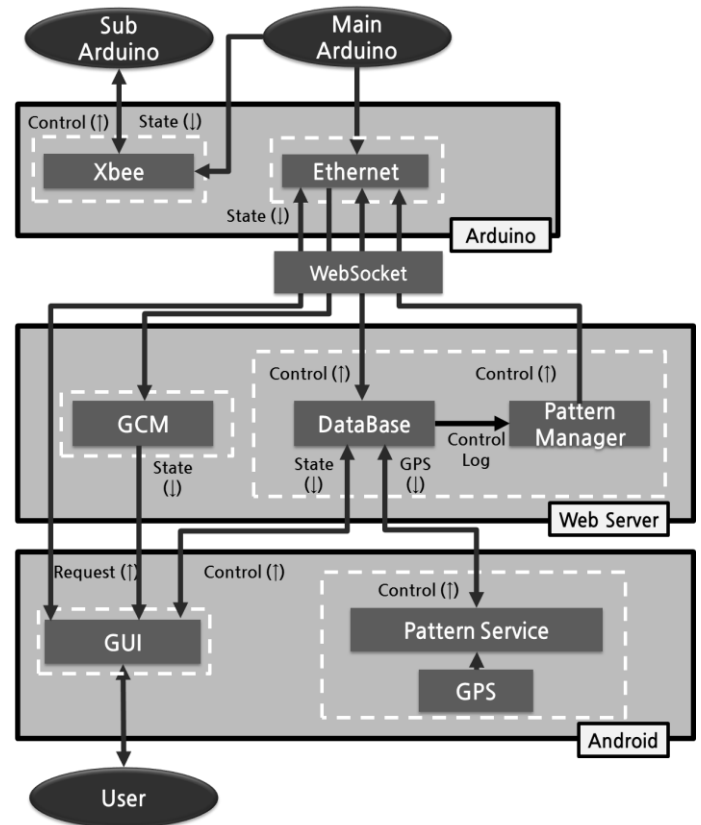


그림 1 시스템 구조

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 서울어코드
활성화지원사업(IITP-2016-R0613-16-1150)의 연구결과로 수
행되었음

2.1 안드로이드 스마트폰

1) 사물 제어 : 스마트홈 어플리케이션(이하 앱) UI를 통해 제어하려는 사물과 명령을 선택하게 되면, 앱에서는 사용자가 선택한 내용에 대해서 데이터 객체를 만들고 HTTP 네트워크 작업을 하는 스레드를 통해 GET방식으로 파이썬 웹서버에 전송을 하도록 하였다.

2) 상태보기 : 상태보기 메뉴를 선택하였을 때, 상태를 보고자하는 사물의 이름을 사물제어에서 사용했던 네트워크 스레드를 통해 마찬가지로 GET방식으로 파이썬 웹서버에 전송하면, XML형식의 상태를 받아온다. HTTP 스레드는 반환된 XML을 파싱을 하여 사물에 대한 상태정보를 얻고, 사용자의 앱에 상태를 보여주도록 하였다.

3) 패턴 제어(GPS) : 서버에 저장된 사용자의 패턴 정보를 앱의 내부 데이터베이스에 저장을 하였다. 그 패턴 정보를 사용자의 앱 화면에 보여주고 자동으로 실행하도록 할 것 인지 선택할 수 있게 하였다. 위치 기반 패턴일 경우, 사용자의 집 위치와 스마트 폰의 현재 정보를 일정한 주기로 계산하여 패턴의 정보(위치, 온도, 시간)차이가 일정 오차 범위 이내라면 서버에 자동으로 제어 요청을 보내도록 하였다.

2.2 파이썬 웹 서버

1) 웹 서버 구축 : 안드로이드 기기, 메인 아두이노와 통신하는 웹 서버는 파이썬 기반의 마이크로 웹 프레임 워크인 Flask를 이용하여 구현하였다. 스마트폰에서 URL을 통하여 서버에 요청을 보내면 서버는 라우팅을 거쳐 해당 URL에 맞는 동작을 수행한다. 제어 명령의 경우 GET방식으로 받은 명령을 DB에 저장하고 데이터를 웹소켓을 통해 메인 아두이노로 전송한다. 상태보기의 경우 사물의 상태를 가져와 XML형식으로 출력한다.

2) 웹소켓 : HTTP 프로토콜은 클라이언트가 요청을 보내고 서버가 응답을 보내는 형식이다. 하지만 본 시스템에서는 요청이 오지 않아도 먼저 메시지를 클라이언트(메인 아두이노)로 전송해야 한다. 따라서 HTML5에 포함 된 웹소켓을 이용하여 통신을 구현하였다. Flask와 함께 구동될 수 있는 Gevent의 웹소켓을 사용하였다.

웹소켓을 통해 주고받게 되는 사물 제어 명령 메시지, 상태 요청 메시지, 상태 정보 메시지를 구분하기 위하여 메시지 프로토콜을 정의하여 통신하였다.

사물(서브 아두이노)이 보내는 상태 정보는 보통의 상태(에어컨 On/Off), 실시간성이 중요한 상태(온도), 긴급성이 있는 상태(창문의 여닫힘)가 있다. 보통의 상태는 단순히 DB에 저장하고 사용자가 상태 정보를 요청하면 가장 최근의 상태를 DB로부터 가져온다. 실시간 상태는

사용자가 상태 정보를 요청하면 메인 아두이노로 상태 요청 메시지를 보내 서브 아두이노로부터 현재의 상태를 받아 와서 출력하는 동시에 DB에 저장한다. 메인 아두이노를 통해 서브 아두이노로부터 긴급 상태가 들어오면 GCM을 통해 사용자에게 푸시를 보낸다.

3) 패턴 알고리즘 : 사용자의 패턴을 분석하여 자동 제어 서비스를 제공하는 기능은 에어컨 제어 기능으로 하였다. 분석하고자 하는 사용자의 패턴은 사용자가 어떤 상태(위치, 시간), 어떤 환경(온도)에서 제어를 많이 하는지를 구하는 것이다.

온도 같은 경우에는 제어를 시작했을 때의 온도가 한 부분에 집중될 것이기 때문에 이 분포를 제어했을 때의 온도의 평균과 표준편차를 이용하여 gaussian distribution으로 가정하였다.

반면에 위치 같은 경우에는 어느 한곳에 집중되어 있기 보다는 여러 부분에 존재할 수 있기 때문에 위치를 포함하는 제어기록을 클러스터링하였다. 이때 클러스터링 알고리즘은 가장 일반적으로 사용하는 k-medians, means 알고리즘이 있는데 outlier의 문제로 전자를 사용하기로 하였다. 또한 여기에 elbow method를 사용하여 가장 적절한 클러스터의 개수를 정하였다. 패턴은 클러스터의 median값으로 하였는데, 이때의 분산, 데이터 개수를 threshold하여 클러스터링 된 데이터들을 다시 한번 구분해주어 좀 더 정확한 패턴을 구분하였다.

2.3 메인 아두이노

1) Setup : 메인 아두이노는 웹 서버의 클라이언트이다. 이더넷 쉴드를 통해 인터넷에 연결하여 웹 서버와 통신할 수 있게 하였다. 아두이노가 켜지면 이더넷 연결을 시작하고, 서버의 웹소켓에 연결을 시작한다. 웹소켓 이벤트를 정의하여 웹소켓으로부터 데이터가 들어오면 데이터를 서브 아두이노로 전송한다. 또한 서브 아두이노로부터 받은 데이터를 웹소켓을 통해 서버로 전송한다.

2) Xbee 통신 : 아두이노 간의 통신 방법은 IEEE 802.15.4 프로토콜을 기반으로한 Xbee 근거리 통신방법을 채택하였다. 각 Xbee들을 중앙 아두이노를 기준으로 Star 형태의 네트워크 통신을 하도록 설정하였다.

Xbee를 통신을 할 때 시리얼을 이용하여 통신을 하고, 여기에도 소프트웨어 통신과 하드웨어 통신 방법이 있다. 본 개발에서는 하드웨어 방식을 채택하였다. 이는 소프트웨어 방식으로 통신을 하게 되면 아두이노 시스템의 자원을 할당해야 해서 통신을 하기 때문에 오버헤드가 크고 안정성과 효율성이 떨어지기 때문이다.

2.4 서브 아두이노

서브 아두이노는 메인 아두이노가 보낸 명령을 받아서

실제로 사물을 제어한다. 사물의 상태가 바뀌면 사물의 종류에 따라 보통의 사물이면 보통 상태 메시지를, 긴급성이 있는 사물이면 긴급 메시지를 Xbee 통신으로 메인 아두이노에게 보낸다. 실시간 상태의 경우 상태 정보 요청이 들어올 때 메시지를 보낸다.

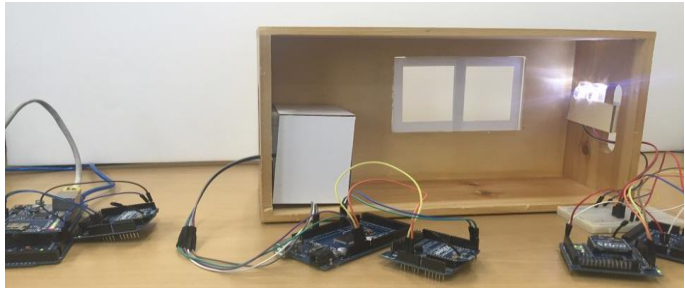


그림 2 스마트 홈 모델

3. 시스템 성능 검증

원활한 네트워크 환경에서 웹 서버와 웹소켓의 상태가 정상적일 때 테스트를 진행하였다.

3.1 사물 제어 및 상태보기

표 1 테스트 결과

	구분	평균 소요시간	수행 확률
사물 제어	1m	2.60 sec	100%
	5m	2.61 sec	100%
	10m	2.67 sec	99%
상태 보기	기본	1.63 sec	100%
	실시간	4.89 sec	98%
	긴급	4.37 sec	100%

거리 별로 제어 명령을 보내어 사물제어가 몇 퍼센트의 확률로 수행되는지, 수행 시간은 평균 몇 초인지를 테스트하였다. 그 결과는 위와 같다.

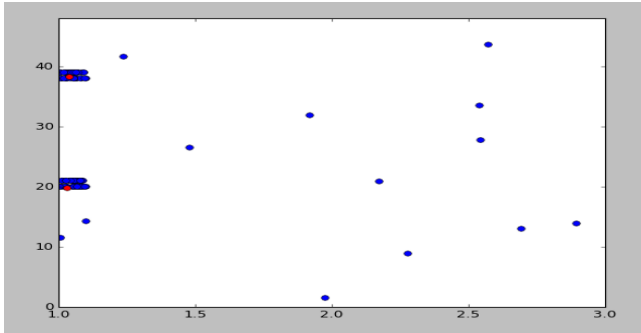
기본(에어컨, 조명 ON/OFF상태), 실시간(온도), 긴급(창문의 열림/닫힘 여부) 상태 보기를 실행하여 사물의 상태를 화면에 띄우는 데에 걸리는 시간이 평균 몇 초인지를 테스트하였다. 그 결과는 위와 같다.

기본 상태의 경우 데이터 베이스에서 바로 상태를 가져오는 것이므로 시간이 제일 적게 소요되었고 실시간

상태의 경우 라운드트립 통신이 이루어지므로 상대적으로 시간이 많이 소요되었다. 긴급 메시지 역시 GCM을 통해 푸시 알림을 보내는 데에 시간이 소요되었다.

3.3 패턴 분석

표 3 패턴 제어 테스트



90%의 사용자의 일정한 패턴을 갖는 데이터와 10%의 outlier를 랜덤으로 넣었을 때 K Medians 클러스터링을 사용하여, 사용자의 패턴(붉은 점)이 제대로 출력되는 것을 확인하였다.

4. 결 론

본 시스템은 사용자가 원거리에서 스마트폰을 이용하여 집안의 사물들을 제어하고 사물들의 상태를 알 수 있는 스마트홈 시스템이다. 더 나아가 사용자의 데이터를 저장하고 이를 기반으로 행동 패턴을 분석하여 자동 제어 서비스를 제공한다.

이렇듯 저렴한 아두이노를 통해 경제적인 스마트홈 시스템을 구현하였고, 사용자의 행동 패턴을 분석함으로써 사용자에게 편리한 서비스를 제공한다.

5. 참고 문헌

[1] 글로벌 스마트홈 경쟁 서막 열리다, http://navercast.naver.com/magazine_contents.nhn?rid=2598&contents_id=98259
[2] IEEE 표준 국제전기전자기술자협회, <http://standards.ieee.org/findstds/standard/802.15.4-2015.html>
[3] Kaazing, <http://www.websocket.org/>
[4] Links2004, <https://github.com/Links2004/arduinoWebSockets>
[5] Python Software Foundation, <https://pypi.python.org/pypi/gevent-websocket>
[6] 김동혁, K-medians 클러스터링과 Elbow method를 이용하여 정확도가 향상된 Snort rule classifier, pp.7~15, 2016