

LoRa IoT 환경망 구축 및 AI 스피커를 이용한 LoRa 모듈 제어

LoRa IoT network environment construction and LoRa module control through AI speaker

요 약

LoRa Alliance™ 에서 제안하는 LoRaWAN network architecture(LoRaWAN 환경망)을 구축하고 AI Speaker를 이용하여 LoRaWAN IoT 환경망 안의 LoRa 모듈들을 제어하는 시스템이다. 집안의 가정으로 한정, 집의 AP와 LoRa Backhaul을 연결하여 게이트웨이(gateway) 역할을 수행하며 집 안의 LoRa 모듈이 장착된 아두이노(Arduino)들을 노드(Node)라고 칭하여 메쉬(Mesh) 네트워크 망을 구성하여 특정 노드들 에게 명령어를 하달, 특정 노드만 실행하는 환경망 및 모듈 제어를 제시 한다. 본 연구에는 LoRa모듈과 아두이노 그리고 서보모터(servo)를 사용하여 가전 스위치에 불임으로써 AI Speaker 와 Web 에서의 집의 전등을 제어한다는 상황을 예시로 한다.

1. 서론

최근 발표된 미래창조과학부에서 조사된 자료에 의하면 2013년 대비 2020년까지의 IoT 시장의 규모는 국내 기준으로 10.4배 정도의 확장을 예측하고 있는 자료를 내놓았다. AI의 비 지도 학습 모델의 지속적인 논문 발표와 GAN 모델의 등장으로의 간단한 분류, 인식 작업뿐만이 아니라 Generative의 역할을 하고 있는 등 IoT와 AI의 입지성은 점점 커져가고 있다.

그중 LoRa Alliance™ 에서 IoT 제품들을 고유 상황이나 국가의 지형지물을 고려하여 적당한 주파수대역을 표준으로 정하는 움직임이 있었다. 따라서 한국에서는 880Mhz ~ 927.5Mhz 정도의 저주파대역을 선택하여 연구, 개발중이다. SKT는 LoRa Module인 (LOM102A)을 시작으로 전국적으로 기지국을 설치 완료 하였고, LOM102A 모듈을 릴리스(release)하여 스타터킷(starter Kit)으로 제공중이다. 한국 표준 주파수인 900Mhz의 주파수를 사용할 수 있는 LoRa 모듈칩을 이용해서 LoRa IoT network architecture를 만족하는 환경망을 구성한다.

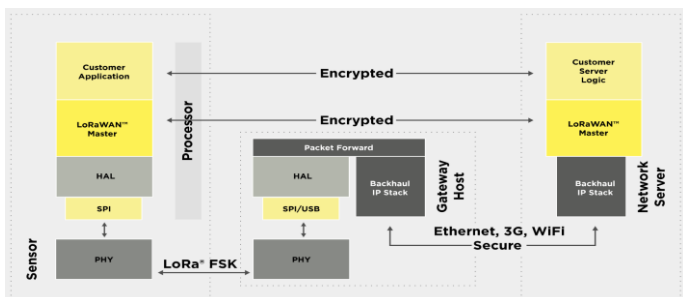


그림 1 LoRaWAN Network Architecture

외부에서는 AP위에 올라간 웹 서버를 이용 하여서 집안 LoRa 모듈에 제어가 가능하며, 내부에서는 AI스피커(Speaker)를 사용하여 집안의 LoRa 모듈을

제어 할 수 있는 연구이다. [그림 1]에서 제시된 AP에 무선으로 연결된 Backhaul의 역할은 LoRa모듈과 Esp8266 메인보드와의 결합을 통해 게이트웨이처럼 AP를 통해 들어온 명령을 토대로 LoRa 모듈의 Send부분의 App key나 Address 값을 메시지에 실어서 보낸다. Sensor의 역할을 할 LoRa 모듈(이하 노드)는 LoRa 칩과 아두이노(Arduino) 와 서보모터(Servo)의 결합으로 신호의 해석으로 물리적인 힘을 내어서 특정 스위치의 압력을 주어 On/Off 기능을 한다.

게이트웨이에서 특정 LoRa 모듈만 신호를 받게 하거나, 모든 신호를 broadcast를 하되 메시지를 열람 후 특정 LoRa 모듈만 실행을 하는 식으로 진행, 사용자가 집안의 IoT 환경망 구축 및 AI스피커를 통해 제어를 하는 것이 연구의 목표이다.

게이트웨이나 노드에서의 날라온 신호들은 AES128의 암호화를 거쳐 LoRa칩에서 복호화 되며 전파법 주기에 의거 5ms 이상의 딜레이를 적용시켜 연구환경을 설정하였다.

2. 관련 연구

현재 시중에 판매 중인 앱으로 제어하며 스위치에 탈부착 가능한 ‘스위치’ 라는 제품과 2017년 SKT IoT 메이커톤 수상작 중의 하나인 LoLock 그리고 ‘샤오미의 Smart Home Kit’를 들 수 있다.

‘LoLock’는 실제 제공되는 LoRa 키트를 이용 SKT 본사에서 진행되었으며 SKT LoRa 망이 구축된 상태로 라라셀드에 아두이노와 블루투스 센서를 부착하여 다가움을 감지하며, 자동으로 도어락이 열리고 일회용 키 발급을 통하여 일회성 출입을 허용하는 연구이다. 기존 클라이언트가 쓰는 방식은 안드로이드 앱을 이용하여 조작을 가능하게 하였다. LoRa 모듈의

Thingplug를 이용하여 기존 인터넷 망의 고정IP를 통하여 ThingPlug 서버에 전달하면 ThingPlug가 미리 발급된 LoRa칩의 고유번호를 찾아가 그 LoRa칩에 신호를 전달하는 식으로 구성하였다.

‘스위치’는 내부의 정확한 구성은 모르지만 스마트폰의 앱을 통하여 미리 부착해 둔 스위치 위의 버튼을 세개의 막대기가 나와서 누르는 제품이다. 1구 기준으로 57,000원의 가격대를 형성하고 있으며 부착한 본체는 안드로이드 5핀으로 충전 가능하다. 기존의 WiFi 나 이더넷의 연결이 필요하며 내부의 특정 공유기 망이 형성되어야 사용가능 하다는 점이 있고 또한 모터의 위치 또한 조절이 불가능하다. 앱의 기본 제공 기능 빼고는 커스텀이 불가능한 점이 있다.

‘샤오미의 Smart Home Kit’는 외부에서의 접속 가능한 서비스(앱)를 제공하여 기존 사용하고 있는 가구에 붙이면 IoT 제품으로 만들어주는 Smart Home Kit 이다. 처음 시작 시 자체적으로 가지고있는 AP를 연결하여 내가 사용하려는 집의 AP의 정보를 알고 있는 Device와 처음으로 연결을 시킨다. 그후 AP의 정보를 Home Kit 에게 넘겨 setup을 시키는 방식이다. 그후의 외부 접속은 집에 있는 인터넷 망을 통하여 접속하는 식으로 한다.

3. 제안하는 시스템

본 연구에서는 [그림 1]에 해당하는 게이트웨이 및 센서 부분의 직접 제작과, Google Assistant의 STT 라이브러리를 이용한 음성 인식을 바탕으로 나온 텍스트를 토큰나이징하여 나온 형태소들에 대해 N-Gram 알고리즘을 사용하여 분석, 사용자의 의도를 분류하여 게이트웨이에 Request를 보내면 노드들에 신호를 전달하는 환경망 및 AI Speaker를 제안 한다. IoT 모듈은 SKT에서 제공하는 LoRa Module(LOM102A) 칩과, 동일 주파수대를 이용 900Mhz대역의 채널들을 이용해서 저전력, 저비용으로 기존 연구와의 차별점을 둘 수 있으며 향후 발전가능성인 부분의 일부인 스위치 제어 부분을 예시를 들어 실현가능성을 검증하였다.

3.1 시스템 클래스 다이어그램

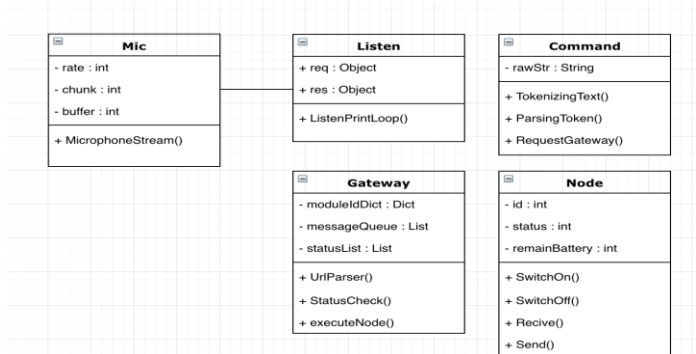


그림 2 클래스 다이어그램

AI 스피커 부분의 Mic와 Listen부분에서의 추출해온 음성을 토대로 토큰나이징을 하여 GateWay에 명령을 내리고, 명령을 받은 Node들에서 실행하는 식의 각각의 필요한 클래스 다이어그램이다.

3.2 시스템 시퀀스 다이어그램

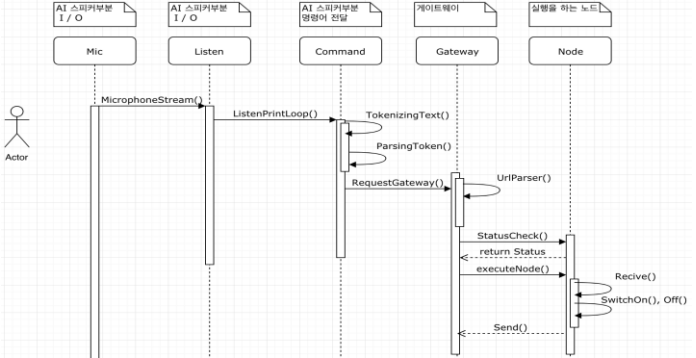


그림 3 시퀀스 다이어그램

3.3 시스템 환경

Backhaul(LoRa 게이트웨이): Wemos D1, esp8266 WiFi칩과, LoRa Shield의 결합으로 게이트웨이 역할을 한다. AP와의 SSID로의 설정을 해놓는다.

노드(실제 동작부분): Arduino Uno와 LoRa Shield그리고 n개의 서보모터의 연결을 통하여 특정 ID값을 지닌다. 브로드캐스트된 메시지의 ID값을 확인후에 자기 ID와의 동일 시에만 작동한다.

AI스피커: Raspberry Pi + Mic 조합 또는, 일반 노트북 + Mic의 조합을 이용하여 STT 오픈소스인 구글 어시스턴트를 이용 말한 내용을 Text로 변환하여서 토큰나이징을 거친 후 N-Gram 기법으로 품사들을 태깅하여 원하는 행동, 원하는 기기를 추출하여 pre-define된 URL로 Request를 보내는 식으로 구성한다.

3.4 선행 연구와의 차이점 및 해결방안

기존 연구의 문제점으로는 크게 외부에서의 접속이 안되는 블루투스 또는 직접사용 방식을 취했다는 점이다. 또한 한대의 노드의 추가가 있을 시에의 추가비용은 기존의 노드가격과는 똑같고 서로의 동작 성공 여부의 Status를 받아서 처리할 중간의 네트워크 망이나 게이트웨이의 부재로 인해서 정확히 실행의 성공 여부나 신호의 전달 여부를 모른다. 또한 전력문제로도의 한번의 완충으로 인해 사용기간이 1주를 넘기기 힘들다는 점이다.

이에 대해 차이점을 두고 해결할 수 있는 방안으로는 LoRa모듈을 탑재한 Kit를 개발하기 위한 외부에 연결된 게이트웨이 또는 백홀로서 LoRa + 라즈베리파이를 사용 Data Link 계층으로부터 프로토콜을 직접 제작함으로 저비용, 저전력 서버를 구성할 것이다. 또한 웹 서버를 제공하여 웹 사이트에서 명령 성공 여부나 동작 여부 등을 실시간으로 확인할 수 있는 플랫폼을 제공한다.

4. 개발 내용

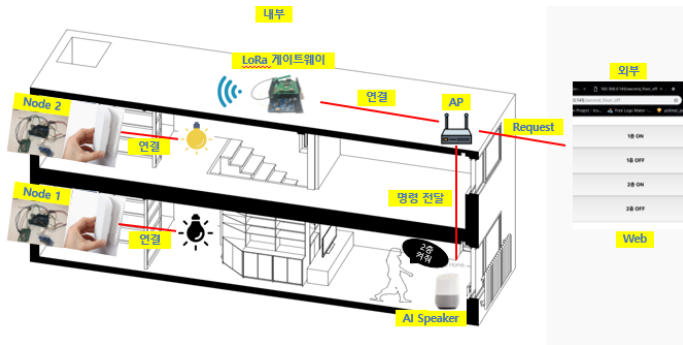


그림 5 상황 도식도

[그림 5]에 보이는 본 연구에서 원하는 상황과 흐름에 대해서 정리한 그림이다. 사용자가 AI스피커에게 특정 음성으로 명령을 내린다.

```
(yhleeEnvPy3) baghyeongjun-ui-MacBook-Pro-2:2018-capston2 phj$ lp
python AI_Speaker_By_Mic.py
/Users/phj/Documents/VirtualEnvs/yhleeEnvPy3/lib/python3.6/site-packages/konlpy/tag/_okt.py:16: User
Warning: "Twitter" has changed to "Okt" since KoNLPy v0.4.5.
warn("Twitter" has changed to "Okt" since KoNLPy v0.4.5.")
/Users/phj/Documents/VirtualEnvs/yhleeEnvPy3/lib/python3.6/site-packages/google/auth/_default.py:66:
UserWarning: Your application has authenticated using end user credentials from Google Cloud SDK. W
e recommend that most server applications use service accounts instead. If your application continue
s to use end user credentials from Cloud SDK, you might receive a "quota exceeded" or "API not enabl
ed" error. For more information about service accounts, see https://cloud.google.com/docs/authentica
tion/.
warnings.warn(CLOUD_SDK_CREDENTIALS_WARNING)
나 : 마이크 테스트입니다
>>> 전체 형태소 : [['(마이크', 'Noun'), ('테스트', 'Noun'), ('이다', 'Adjective')]
>>> 명사(Noun) 형태소 : ('마이크', 'Noun'), ('테스트', 'Noun'),
>>> 숫자(Number) 형태소 :
>>> 동사(Verb) 형태소 :
>>> Weight 값 : Noun = -1, Number = -1, Verb = -1
>>> GET : 분석 실패하여 Log 를 저장합니다.

나 : 1층전동거기
>>> 전체 형태소 : [['1', 'Number'), ('층', 'Noun'), ('전동', 'Noun'), ('거기', 'Verb')]
>>> 명사(Noun) 형태소 : ('층', 'Noun'), ('전동', 'Noun'),
>>> 숫자(Number) 형태소 : ('1', 'Number'),
>>> 동사(Verb) 형태소 : ('거기', 'Verb'),
>>> Weight 값 : Noun = -1, Number = 0, Verb = 1
>>> GET : http://192.168.0.145/first_floor-off
```

그림 6 Text토큰나이징 화면

[그림 6]에 AI스피커는 STT가 되어 넘어온 텍스트들을 토큰나이징하여 3어절 단위로 분석하여서 사용자의 의도를 미리 맵핑된 행동양식에 따라서 URL을 AP에 요청한다. 요청된 AP의 포트포워딩된 URL을 LoRa백홀(게이트웨이)가 수신하여 URL Parsing을 시작한다. URL파싱을 하여 LoRa Send부분의 address값을 변경해 신호를 Send하면 Receiver에서는 브로드캐스트된 메시지를 각각 열람하여서 자기의 ID값과 맞으면 자기와 연결된 서버모터를 작동한다. 특정 스위치나 여러 가전제품을 제어하는 부분에 붙일수 있게끔 DIY하여 사용할 수 있다. 게이트웨이와 노드 간의 메쉬네트워크 구성을 통하여 게이트웨이와 센서모드들다 send와 receiver 반복하여 특정 신호를 보내어 받고 처리를 하고 처리한 결과를 다시 send하는 과정을 통해 Backhaul이 중앙관리하여 AP에 업로드 된 웹 서버에 지금 노드들의 정상작동여부를 확인할 수 있으며 방금 보낸 메시지큐에대한 동작성공여부등을 실시간으로 관리할수있다. 내부에서는 AI스피커 외부에서는 웹 플랫폼의 제공으로 인해 기존 문제점의 한계도 해결하면서 저전력, 저비용, 간편함을 구현하였다.

5. 결론 및 향후 계획

이 연구에서는 LoRa 모듈을 사용하여서 IoT 인터넷망을 집안에 구축해보고 AI 스피커를 통해서 IoT 환경망에 접근 LoRa 모듈이 포함된 노드들을 직접 제어하는 부분을 모두 해보았다. 900Mhz 라는 저주파수대역대의 간섭 및 굴절이나 뻗어나가는 속력 문제등을 해결 하기 위해 메쉬네트워크를 구성하여서 실제의 동작 여부를 확인하는 점과, 노드 역할을 하는 부분에는 App key 나 ID 값의 도입으로 브로드캐스트만 될수있는 로라모듈칩의 한계를 다른 식으로 우회하여 풀어나갔다. LoRa 모듈의 physical layer 부분의 HAL 레이어단의 아래부분에서의 직접적인 제어는 안되었지만 그 윗부분의 TCP/IP 의 통신 부분 및 AES 의 암호화 부분등의 전면 수정이 가능하다는 것을 알게 되었고 실제로 서의 사용시에는 신호의 간섭과 그 노드가 신호를 받고 처리하고나서의 Send 부분의 꼭 필요함을 알게 되었다. 지금은 모든 노드들이 게이트웨이의 부분을 바라보며 한 방향 집중방식으로 네트워크를 구성해보았지만 노드와 노드끼리도 연결되어있고 최종의 결과를 내는 부분은 게이트웨이의 역할을 두는 것도 성능을 향상시키는 방법 중에 하나였다.

향후 계획으로는 응답 속도의 시간을 줄여보는 연구를 진행해볼 것이며 그 부분에서의 LoRa 모듈의 쓰는 채널의 8 채널중 적합한 채널과 거리에 따른 신호 세기를 조절하여서 SF7~12 사이의 적합한 값을 찾는 알고리즘을 세워서 최적의 신호 강도 및 채널을 찾아 세팅해주는 연구를 해서 도입해보고 싶다. 또한 간섭은 충분히 네트워크망으로도 해결할 수 있으나 서로의 상태체크부분에서의 너무 많은 시간을 소요 함으로 성능에 저하를 할 수도 있어서 이 부분의 최적화를 하는 알고리즘에 대한 연구를 해볼 계획이다. 연구 초반에 계획을 잡았던 저전력이며, 저비용의 IoT 환경망을 구축하고 기존의 IoT 제품이 아니었던 가전제품을 IoT 제품으로 만들어주는 Kit 의 개발에도 기여할 수 있다고 생각하는 바이다.

참 고 문 헌

- [1] SKT. "LoRa 장치 개발 가이드". 『SKT 』, 2016. 04
- [2] 박형준, "컨셉영상", https://www.youtube.com/watch?v=FtFO_utAEcE&t=106s
- [3] 박형준, "스위치 제어 개념설명영상", https://www.youtube.com/watch?v=FtFO_utAEcE&t=106s
- [4] 박형준, "실제 작동영상", https://www.youtube.com/watch?v=FtFO_utAEcE&t=106s