

Table of contents

1	Wireless IoT를 위한 RFLink RF gateway	2
1.1	왜 433Mhz일까?	2
1.2	RFLink gateway	3
1.3	펌웨어 업로드	4
1.4	프로토콜	4
1.5	Play with Node-red	4

1 Wireless IoT를 위한 RFLink RF gateway

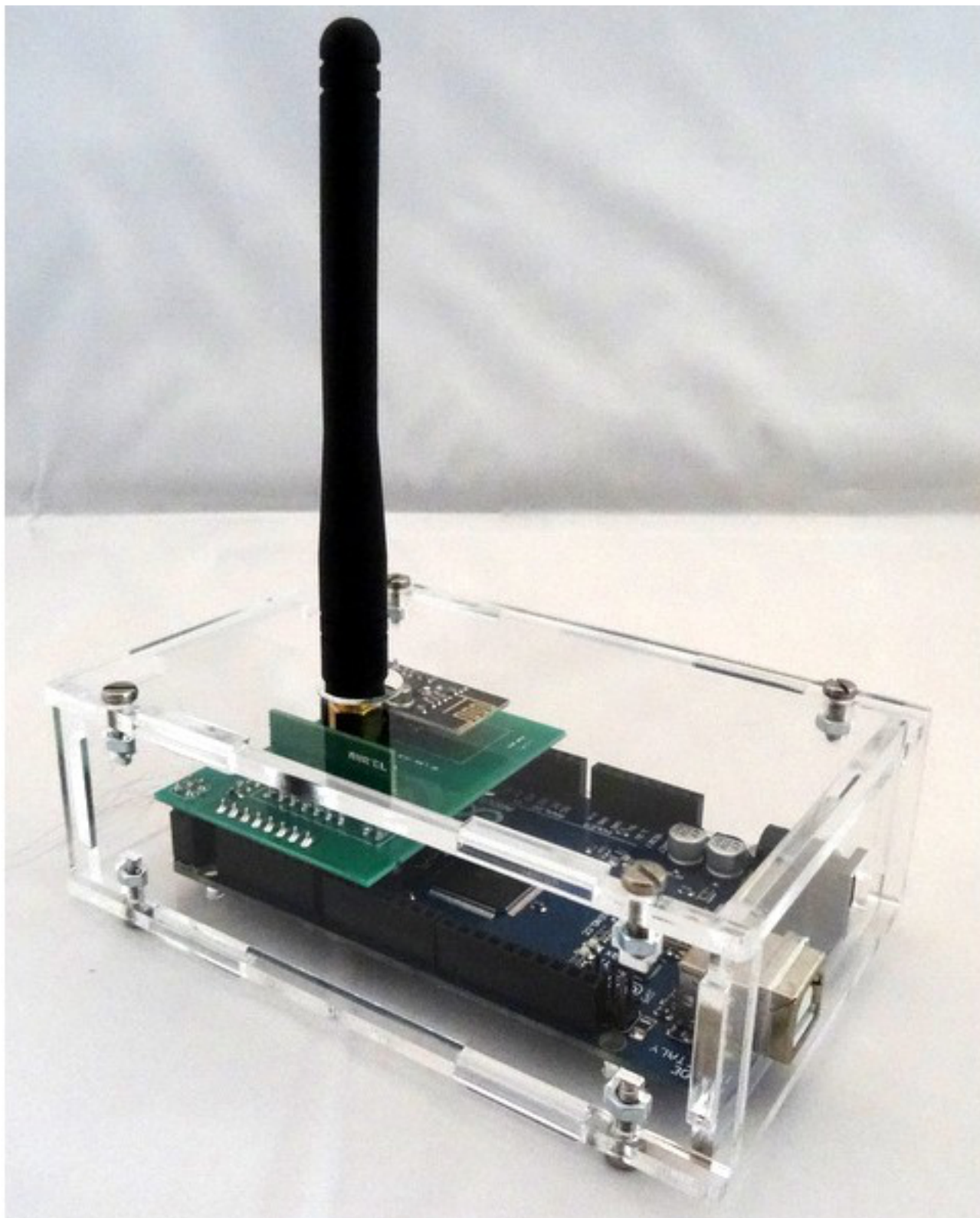
1.1 왜 433Mhz일까?

433Mhz는 라이선스가 필요없다. 이말인 즉슨, 누구나 이 주파수를 사용하는 기기를 아무 제약없이 만들 수 있다는 점이다. 이러한 이유로 인해 433Mhz를 사용하는 수많은 장비들을 찾아볼 수 있다. 온/습도계, 화재경보기, 도어벨, 차고 개폐, 창문/문 개폐 감지 센서, PIR 센서 등 정말 다양한 장치들을 쉽게 구할 수 있다. 그런데, 다양하다는 것은 엉망진창이기도 하다. 단일의 프로토콜이 없어서, 각 기기가 다른 방식으로 통신을 한다. 보안성도 없어서 누구나 신호를 받을 수 있다.

현 시점에서 RF의 경쟁자는 BLE와 Zigbee라고 생각된다. ZWave가 있지만, 국가간 주파수 대역이 달라서 저렴한 제품을 다른 나라에서 구하기가 어렵고 기기 자체의 수나 제조업체의 수가 그다지 많지 않은 거 같다. Wifi의 경우 조금씩 많아지지만, 배터리로 운영되는 게 편한 온/습도계, 화재 경보기, 개폐 감지 센서, PIR 센서 등 센서류에는 배터리 수명 문제가 있을 것 같다. BLE는 점점 기기수가 많아지는 거 같고, Zigbee 또한 Xiaomi Smart Home suite 덕분에 조금은 저렴하게 구할 수 있게 된 거 같다.

개인적으로 전원선을 연결하기 번거로운 센서류는 RF 장치 사용이 편하고 저렴하게 구할 수 있어서, [RFLink gateway 모듈 + 케이스](#) + 호환 아두이노 메가 2560, 그리고 [Acurite 00592TXR 온/습도계](#)를 구입했다.

1.2 RFLink gateway



RFLink gateway는 RF 디바이스들을 위한 gateway이다. 주로 433Mhz를 사용하는 기기들을 지원하지만 이외에도 315Mhz, 866Mhz, 915Mhz 및 2.4Ghz 사용하는 몇몇 디바이스들도 지원해준다. RF signal을 송/수신 할 수 있으므로, 433Mhz 기반의 온습도계의 데이터를 읽거나, 433Mhz 기반의 AC switch를 제어하는 데 사용할 수 있다.

상용 제품으로는 RFXCom이라는 잘 만들어진 제품이 있으나 가격이 좀 비싸다. 반면에 RFLink gateway는 Arduino Mega + 호환 RF transmitter/receiver를 가지고 직접 만들수도 있고, nodo-shop.nl에서 판매하는 모듈을 사서 끼워도 되므로 RFX-com에 비해 저렴하게 구현할 수 있다. ESP8266 기반으로 RF 송수신 모듈을 연결해서 구현하는 것보다는 비싸지만, 많은 기기를 지원(지원 장치 목록)하는 SW가 지원된다는 점이 강력한 장점이다. 다른 RF 관련한 오픈소스 프로젝트 RCSwitch(주로 RF Switch류 모니터링/제어용)하고 rtl_433(수신만 가능)하고도 비교해봤지만, 기기 지원면에서 RFLink gateway가 가장 좋은

것으로 판단되었다. 또 하나의 이유로는, nodo-shop.nl에서 모듈과 Arduino Mega 2560을 같이 수납할 수 있는 케이스도 판매하고 있어서 깔끔하게 사용이 가능하다는 것도 강력한 이유였다. 개인적으로 PCB 보드들이 주렁주렁 선으로 연결되어 있는 방식의 DIY를 좋아하지 않기도 했다.

RFLink gateway는 원래 Open source project였으나, 어떤 이유에서인지 Freeware project로 변경이 되었다. 즉, 소스 자체는 공개되어 있지 않지만 사용에는 제약이 없는 형태로 운영이 되어지고 있다. 새로운 기기 지원을 위해서는 Plugin 개발을 해야하는 데 이 경우 운영자에게 직접 문의해서 진행하는 방식인 듯 하다. 다만 Freeware project로 변경되지 이전까지의 소스는 다음의 2 곳에서 찾을 수 있었다.

- <https://github.com/ThibG/RFLink>
- <https://github.com/Pirionfr/RFLink-Gateway>

1.3 펌웨어 업로드

펌웨어를 RFLink gateway에 쓰기 위해서는 윈도우환경에서 하는 게 편한데, 업로드 전용툴이 지원되기 때문이다.

- Arduino mega 2560를 윈도우 pc의 usb 포트에 연결한다.
 - CH340G이 사용된 저렴한 3rd party Arduino mega 2560의 경우 윈도우 7에서 만약 드라이버 설치 실패시 [여기](#)에서 다운로드 후 드라이버 업데이트 해야 함
- 장치 관리자에서 시리얼 포트의 속도를 57600-8-N-1으로 설정
- [다운로드 페이지](#)에서 최근 파일 다운로드 후 압축을 해제한다.
- RFLink_loader.exe를 실행해서 포트와 펌웨어를 선택 후 펌웨어 업로드한다.
 - 보드의 led가 빠르게 점멸하다가 멈추면 다운로드가 완료된 것이다. 이 상태에서 만약 RFLinkLoader.exe가 응답없음에 빠지면 그냥 강제 종료하면 된다.
- 이제 가지고 있는 RF 장치를 켜서 패킷이 송신되게 한 후 어떻게 디코드되는 지 확인하면 된다.

1.4 프로토콜

[프로토콜 페이지](#)를 보면 자세히 알 수 있지만, 대략 다음과 같은 구조를 가지고 있다.

수신패킷 20;2D;UPM/Esic;ID=0001;TEMP=00cf;HUM=16;BAT=OK;

- UPM/Esic -- 장치/프로토콜 명칭
- ID=0001 -- 장치 아이디
- TEMP=00cf -- 해당 장치가 내보내는 데이터 중 온도, 00cf = 207 --> 20.7도
- HUM=16 -- 해당 장치가 내보내는 데이터 중 습도, 16 --> 16%
- BAT=OK -- 해당 장치가 내보내는 데이터 중 Battery status

송신패킷 10;AB400D;00004d;1;OFF;

- AB400D -- 장치/프로토콜 명칭
- 00004d -- 장치 아이디
- 1 -- 스위치 번호
- OFF -- 제어 명령

1.5 Play with Node-red

Raspberry Pi 3의 USB 포트에 RFLink gateway를 연결한 후, Node-red에서 Serial node를 생성하면 RFLink gateway가 보내는 데이터를 쉽게 볼 수 있다. 하지만 [해당 메시지들을 손쉽게 가공해주는 전용 node](#)가 없어서 직접 만들었다.