MQTT을 이용한 간이중계소 원격제어시스템

글. 이주용 대구MBC 기술국 방송인프라부 부장 대우

개발 배경

지난해까지 당사에서는 간이중계소의 상태 확인 및 제어에 2세대 이동통신 서비스를 이용해오고 있었다. 별도의 중계소 통합관제 시스템 없이 관리자가 휴대전화 문자 서비스(SMS)를 이용하거나 CDMA 모뎀이 연결된 PC에서 에뮬레이터를 실행하는 방식으로 중계기 원격제어를 실행하였다. 편의상 SMS를 주로 이용하였는데 이러한 방식은 문자와 숫자가 조합된 코드를 송수신하므로 직관성이 떨어질 뿐 아니라, 중계기 이상이 발생하더라도 관리자가 상태요청을 하기 전까지는 이상 여부를 알 수 없다는 문제점이 있었다. 게다가 원격 감시 및 제어가 전혀 불가능한 국소도 있었다.
2020년 2G 종료에 따라 중계소 원격제어 통신망의 변경이 요구되었으며 이동통신 서비스의 변경과 동시에, 기존의 문제점과 불편함을 해소할 수 있는 간이중계소 원격제어시스템을 구축하고자 하였다.

개발 목표

시스템 설계 및 구현에 있어 다음과 같은 사항을 목표로 하였다.

- 개별 간이중계소들을 통합적으로 관리할 수 있는 시스템
- 옥외형 중계기 본체 내부에 설치 가능한 크기의 하드웨어
- 중계기 외의 장비에 대한 상태 확인 및 제어(GPIO)
- 간이중계소의 정전상태 확인
- 사용자 친화적인 인터페이스
- 기존 CDMA 방식과 비슷한 수준의 데이터 통신비용

이상의 사항들을 충족하기 위하여 M2M IoT 기술을 적용, 대응하였으며 당사의 송신소 원격시스템 구축을 함께한 ㈜아크로비젼 사와 협력하여 진행하게 되었다.

시스템 구성

시스템의 주요 구성은 [그림 1]과 같다.

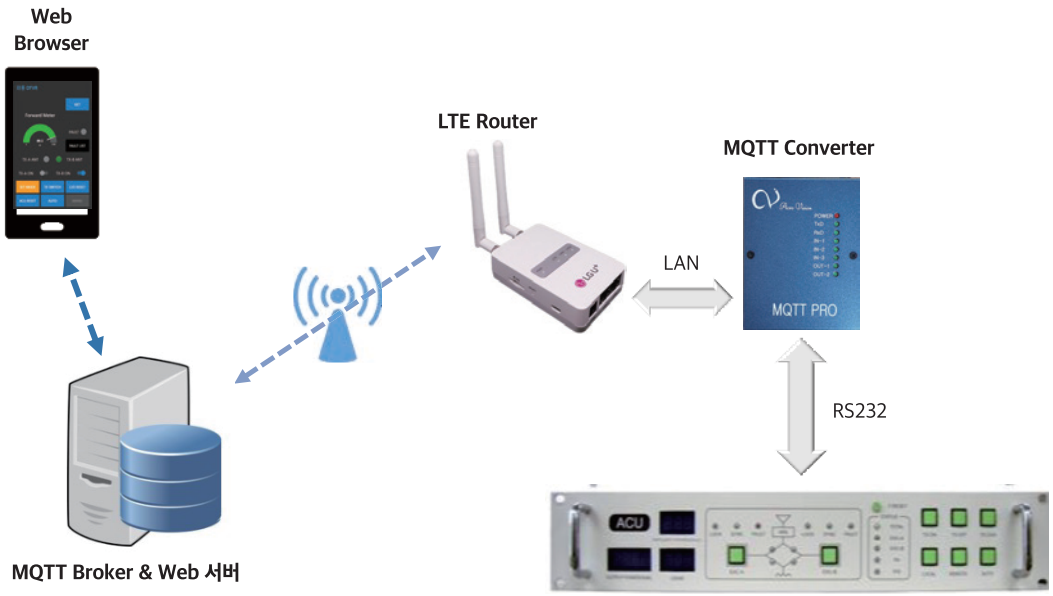

그림 1. 시스템 구성도

이동통신 서비스

2G CDMA의 대체 솔루션으로 통신사에서 제공하는 서비스들을 검토한 결과, LTE Cat.M1과 LTE 라우터(Router)가 가장 적합해 보였다.

Cat.M1은 국제 표준화 단체 3GPP에서 표준화한 LPWA(저전력 장거리) 통신기술로 375kbps의 데이터 속도를 가지며 저용량 사진/Voice 전송, 인증, 결제 등 실시간 이동형 서비스에 적합하다. 현재 우리나라에서는 SKT가 상용화하여 서비스하고 있다.

LTE 라우터는 4G망을 이용하는 무선 인터넷 장비로 통신 3사에서 모두 제공하고 있으며 CCTV, 상점의 키오스크(KIOSK), 카드 결제기 등에 널리 사용되고 있다.

이상의 두 가지 통신방식에 대하여 다음에서 설명할 MQTT 프로토콜의 적용 가능성을 검토하였다. 국내 제조사의 Cat.M1 모뎀 3개 제품에 대한 테스트 결과, 1개 제품만이 MQTT 통신이 가능하였다. 반면에 LTE 라우터는 MQTT의 적용에 문제가 없고 시중에 다양한 제품이 출시되어 있다. 사용 환경에 따라 향후 모뎀(라우터)의 변경이 필요할 수 있으며 간이중계소 원격제어 용도로 Cat.M1이 가지는 상대 우위가 크게 없으므로 보다 범용적인 LTE 라우터를 선택하였다.

무선 라우터 서비스는 IP 할당 방식(고정/유동)과 제공 데이터양에 따라 다양한 요금제가 있지만, MQTT 프로토콜을 적용하면 데이터양이 제한된 유동 IP 환경에서도 필요한 기능을 충분히 구현 가능하였으므로 기존의 CDMA 이용요금과 큰 차이가 없는 요금제를 가입하였다.

MQTT

MQTT(Message Queue for Telemetry Transport)는, M2M, IoT 기기와 Gateway의 연동을 위해 정의된 경량 프로토콜로, 최소한의 전력과 패킷량을 가지기에 저 전력, 저 대역폭 환경에서도 운영이 가능하다. 페이스북, 배달의 민

족 등의 애플리케이션에 적용되었다고 알려져 있다. HTTP 등의 서버-클라이언트 구조와 달리 Broker, Publisher, Subscriber 구조로 되어 있는데, Publisher는 메시지를 발행하고 Subscriber는 관심 있는 주제를 구독하는 것을 기본 원칙으로 한다. Publisher와 Subscriber는 모두 Broker에 대한 클라이언트로 작동하며 Broker(중개인)는 발행인과 구독자가 메시지를 주고받을 수 있도록 다리를 놔주는 역할을 한다. 하나 이상의 Publisher와 Subscriber가 Broker에 연결해서 토픽을 발행하거나 구독할 수 있으며 다수의 클라이언트가 하나의 주제를 구독할 수도 있다.

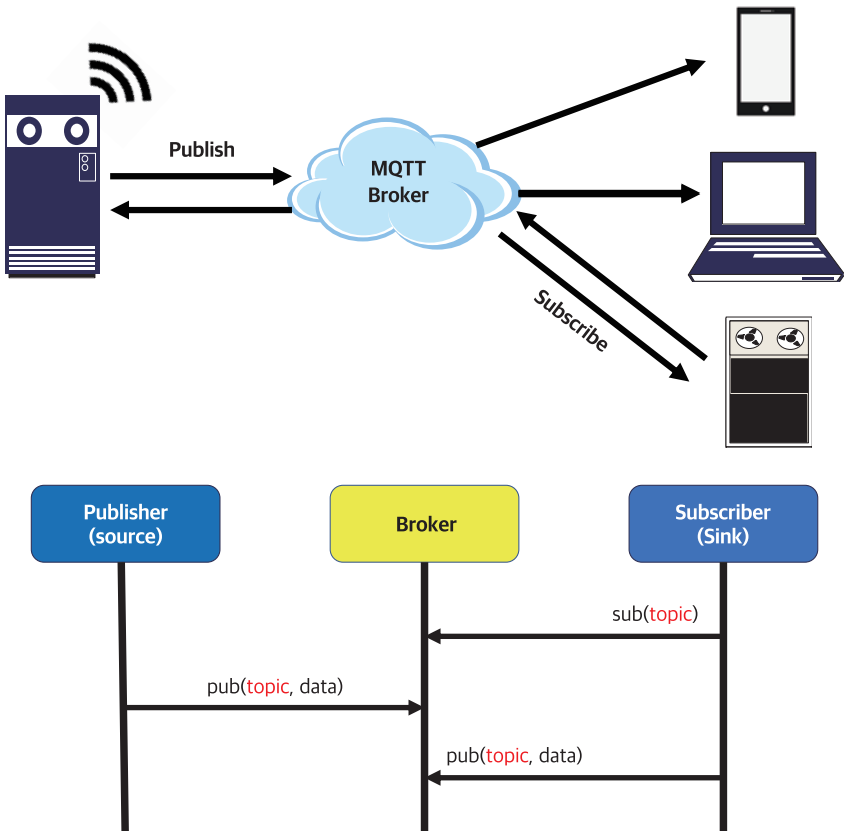

그림 2. MQTT 통신 개념도

MQTT를 이용하는 간이중계소 원격제어시스템은 다음과 같이 작동한다.

간이중계소의 장비, 운영센터의 프로그램 등은 모두 Broker에 대한 클라이언트가 되며 운영 프로그램에서 발생한 제어 명령을 Broker에게 발행(Publish)하면 Broker는 해당 명령을 구독(Subscribe)하는 중계소의 장비에게 메시지를 전송하고, 메시지를 받은 장비는 해당 제어를 실행한다.

반대로 중계소의 장비는 자신의 상태값에 변동이 있을 때, 이를 Broker에게 발행(Publish)하고 운영 프로그램은 Broker를 통해 해당 장비의 상태값을 구독(Subscribe)하여 화면에 표시한다. 이러한 Publish와 Subscribe는 토픽(Topic)을 기본으로 동작하는데, 슬래시(/)로 구분하여 계층적으로 구성할 수 있고 다량의 기기들을 효율적으로 관리할 수 있다. [그림 3]은 당사의 토픽 구성으로 [중계소명] / [방송매체] / [별도동작(Status, Control)]으로 구분하였다.

이러한 방식으로 운영 프로그램은 각 간이중계소 장비와 1:N으로 연결되어 원격 감시제어를 실행하게 된다.

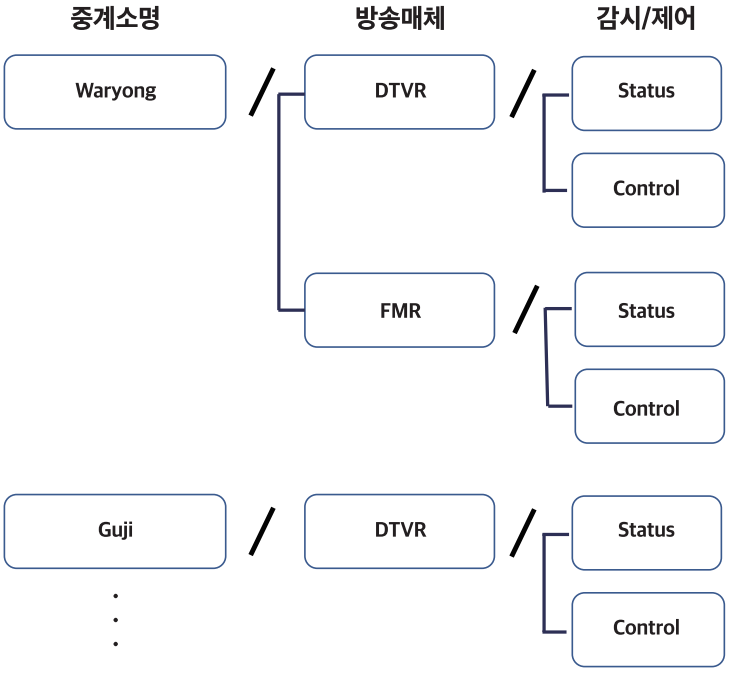

그림 3. Topic 구성도

제어보드 (MQTT 컨버터)

MQTT 컨버터는 Broker와 MQTT 통신을 하는 기능과 함께 중계기를 감시/제어하는 기능을 가진다. MQTT 컨버터는 ACU와 주기적으로 통신하면서 상태를 읽고, 상태에 변화가 있으면 이를 메시지화하여 MQTT Broker로 전달한다. 반면 운영 프로그램에서 발생한 제어 명령은 MQTT Broker를 통해 MQTT 컨버터로 전달되고 이를 해석하여 ACU가 인식할 수 있는 제어방법(시리얼 통신 명령 또는 제어접점 발생)으로 변경, ACU를 제어하게 된다. 이렇듯 MQTT 클라이언트와 프로토콜 컨버터로서의 두 가지 기능이 유기적으로 동작하므로 MQTT 컨버터라 칭하였다.

MQTT 컨버터는 옥외형 중계기 본체 내부에도 설치해야 하므로 보드 사이즈를 최소로 가져가면서 필요 기능만을 구현해야 했다. 처음에는 라즈베리파이를 검토하였으나 확장보드를 붙이기에는 각 포트의 구성이 용이하지 않아 아두이노 모듈을 이용하여 설계하기로 했다. 아두이노는 오픈소스를 기반으로 한 단일보드 마이크로컨트롤러이며, 컴파일된 펌웨어를 USB를 통해 손쉽게 업로드 가능하다. 더욱이 제일 중요한 기능인 이더넷의 TCP/IP와 MQTT 프로토콜이 오픈소스로 제공되기에 개발 기간을 단축할 수 있었다.

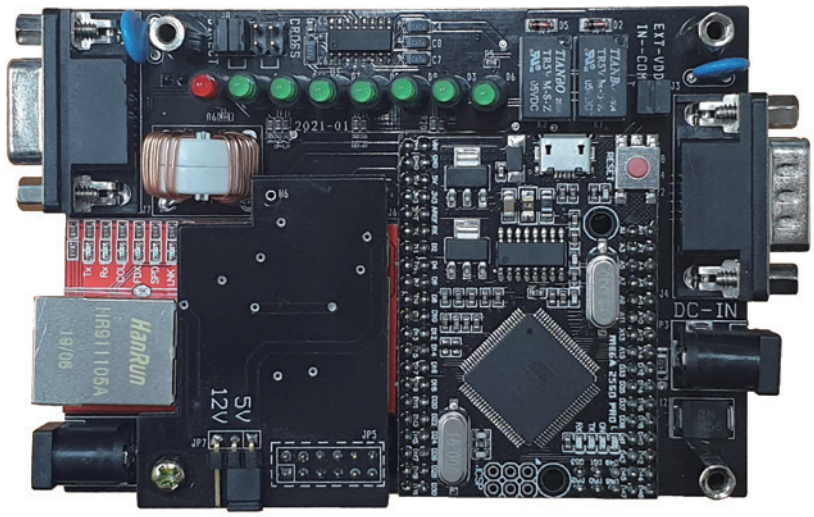

그림 4. 시리얼 통신용 보드

아두이노의 확장보드는 제어하여야 할 중계기 원격 인터페이스에 따라 시리얼 통신용과 접점 제어용, 이상 2가지 형태로 설계했다. 시리얼 통신용 보드의 외형을 보자면 우선 시리얼 통신포트, 이더넷 LAN 포트, 외부 접점 입력(3P) 및 릴레이 출력(2P)을 가진다. 접점 제어용 보드는 시리얼 통신포트 대신 외부 입력접점, 외부 출력접점, 아날로그 입력용 포트를 각각 가진다. 외부 접점 입력은 중계소의 부가장비들에 대한 상태감시 및 제어를 목적으로 주로 냉방장치, 비상발전기의 가동상태 또는 이상 유무를 확인하는데 사용될 수 있다.

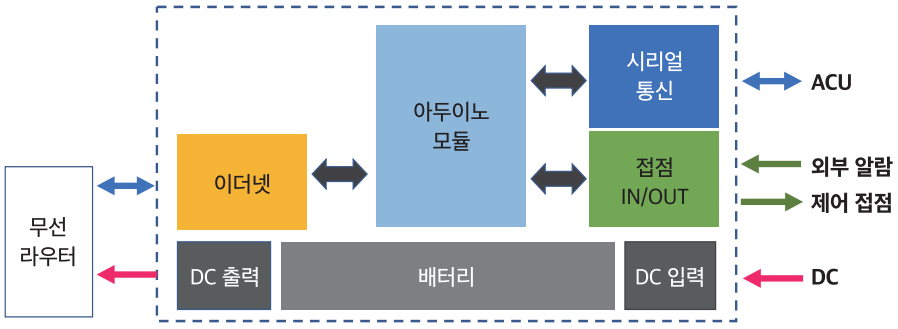

그림 5. 시리얼 통신용 보드 블록도

대다수의 소출력 간이중계소는 예비전원이 없으므로 정전으로 인한 송출중단과 시청 불가 민원이 발생할 경우, 그 원인을 파악하는 데까지 적지 않은 시간이 소요되기도 한다. 이러한 경우를 대비하여 MQTT 컨버터에 배터리를 내장, 외부전원이 차단되더라도 일정 시간 동안 자체 보드와 LTE 라우터에 전원을 공급하여 정전상황을 MQTT Broker에게 알릴 수 있도록 설계하였다.

사용자 인터페이스

운영센터의 원격감시제어프로그램은 Node-RED로 개발하였는데, Node-RED는 IBM이 개발한 시각 프로그래밍을 위한 플로 기반 개발도구이다. 하드웨어 장치들, API, 온라인 서비스를 사물인터넷의 일부로 와이어링 시킬 수 있으며 직관적인 소스 구성이 가능하다는 장점이 있다.

실제 구축은 리눅스(우분투) 서버에 Node-RED와 MQTT Broker를 설치했다. 간이중계소별로 프로그램 단위인 '플로우'를 가지며, 이는 개별 GUI 화면이 된다. [그림 6]은 MQTT 통신 노드와 램프 출력, 그래프, 버튼 등을 와이어링하여 프로그램을 구성한 화면이다.

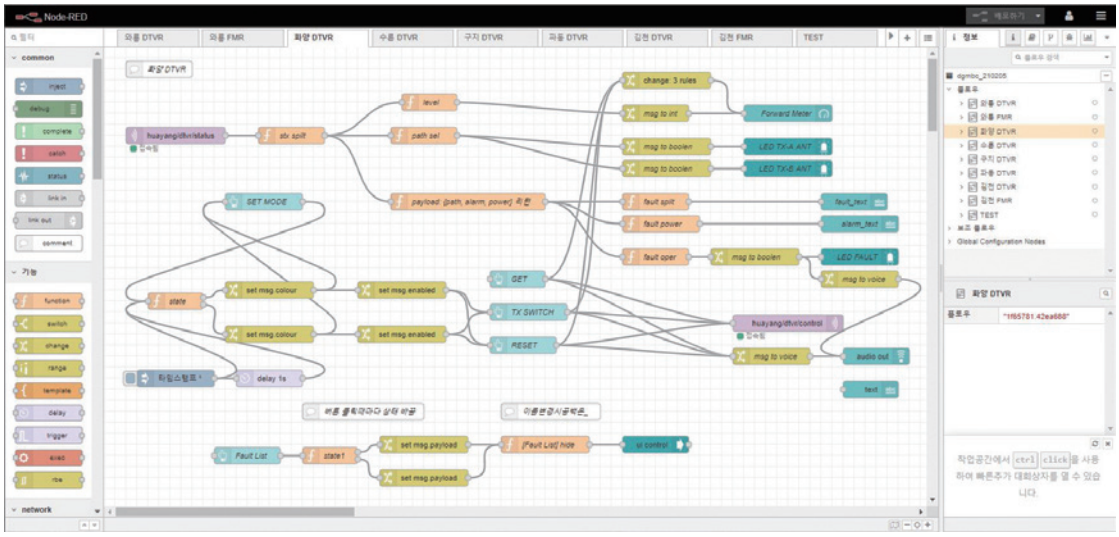

그림 6. Node-RED 플로우 프로그래밍 화면

[그림 7]은 실제 구축된 원격감시제어 UI로, PC뿐 아니라 스마트폰에서도 그대로 사용되고 단말기의 해상도에 맞게 자동으로 조정된다.

그림 7. Node-RED UI 화면(일부)

동작 예를 들자면 [GET] 버튼을 누르면 해당 중계소의 상태를 읽어 와서 상태표시를 갱신한다. 제어를 위해서는 [SET MODE] 버튼을 먼저 눌러 제어 버튼을 활성화한 후 명령을 실행하여야 하는데 이는 스마트폰에서 의도치 않은 접촉

(Touch)에 의한 오작동을 방지하기 위함이다. 제어실행 시에는 일정 시간 후 상태가 자동으로 갱신되어 제어실행 결과를 표시한다.

중계소 중 어느 한 곳에서 Fault가 발생하면 해당 중계소의 Fault LED가 점등되고 주조정실의 상시 모니터링 PC에서 경보음을 발생한다. 세부적인 Fault의 내용은 [Fault List] 버튼을 눌러 확인할 수 있다.

그림 8. 옥외형 중계기 내부 설치

설치 및 운용

서버의 운영 프로그램은 언제든지 수정이 가능하지만 MQTT 컨버터 프로그램 변경을 위해서는 현장을 방문해야 하므로 1개 사이트에서 충분한 테스트를 거친 후 전체 중계소에 적용하였다. 이미 인터넷 회선이 확보되어 있는 중계소는 기존 회선을 이용하고 나머지에 대해서는 LTE 라우터를 개통, 설치하였다. 거의 모든 중계소에서 LTE 라우터는 안테나 다이버시티를 구성하지 않고도 통신이 원활하였으나 1개소는 신호 수신 레벨의 변동이 심하여 옥외 안테나를 설치하여 안정적인 데이터 송수신 환경을 구축하였다. 실제 운용 시 중계기별로 사용하는 프로토콜의 형식과 참조하는 항목의 수에 따라 응답속도에서 약간의 차이가 발생하였다.

마치며

올해 초 새로운 원격제어시스템의 구축이 완료되면서 간이중계소에 대한 상태 확인 및 제어가 한층 용이해졌으며 정기적인 현장점검의 횟수도 줄일 수 있게 되었다. 시스템의 개발 기간 동안 통신방식의 선정, 하드웨어 설계, 프로토콜 해석, UI 디자인 등 다양한 검토를 거쳤으며, 현장의 소소한 문제들과 코드상의 버그도 끊임없이 마주하였다. 하지만 약 2개월간의 짧은 시간 동안 목표로 했던 것들을 대부분 달성하였으며 현재까지 안정적으로 운용되고 있다. 물론, 앞으로도 보완 및 개선이 필요한 부분이 발생할 것으로 생각된다.

마지막으로 설계로부터 구현에 이르기까지 크고 작은 문제들을 해결해 나가는 과정에서 기술적인 협력과 조언을 아끼지 않은 대구MBC 기술국 동료들과 ㈜아크로비젼에 감사의 인사를 전한다.