2018 KBS 미디어창의기술전 소개
IoT 스마트컨트롤러

글. 배인탁 KBS 창원총국 기술국

IoT 스마트컨트롤러 개발 배경

2017년 가을, 지역국 NPS(N/W Production System) 스위치 장비에 장애가 발생한 적이 있었다. 평상시 장비 상태를 원격 모니터해 장애 발생 시점은 즉각 감지했으나, 데이터 네트워크를 통한 원격접속이 불가능해 관리자가 직접 장비를 확인해야 하는 상황이었다. 이를 위해서는 장시간의 복구 시간이 필요했다.

일반적으로 IT 기반 장비는 시스템을 물리적 재기동하는 것으로 문제가 해소되는 경우가 많다. 그래서 작업자가 장애 발생 시 원격지 장비를 물리적으로 재기동하고, 직접 접속해 상태 확인 및 조치가 가능한 환경을 제공해주는 시스템이 필요했다.

IoT 스마트컨트롤러팀 소개

※ **IoT(Internet Of Things):** 인터넷을 기반으로 모든 사물을 연결하여 사람과 사물, 사물과 사물 간의 정보를 상호 소통하는 지능형 기술 및 서비스를 말한다.

IoT 스마트컨트롤러란?

그림 1. 소개 영상 QR 코드 및 URL

IoT 스마트컨트롤러는 원격지 장비들을 통합으로 관제하는 프로그램과 모바일 연동을 통하여 언제 어디서든 원격지 장비의 상태를 확인 및 제어할 수 있는 시스템이다. 제어에는 물리적 전원제어도 포함된다.

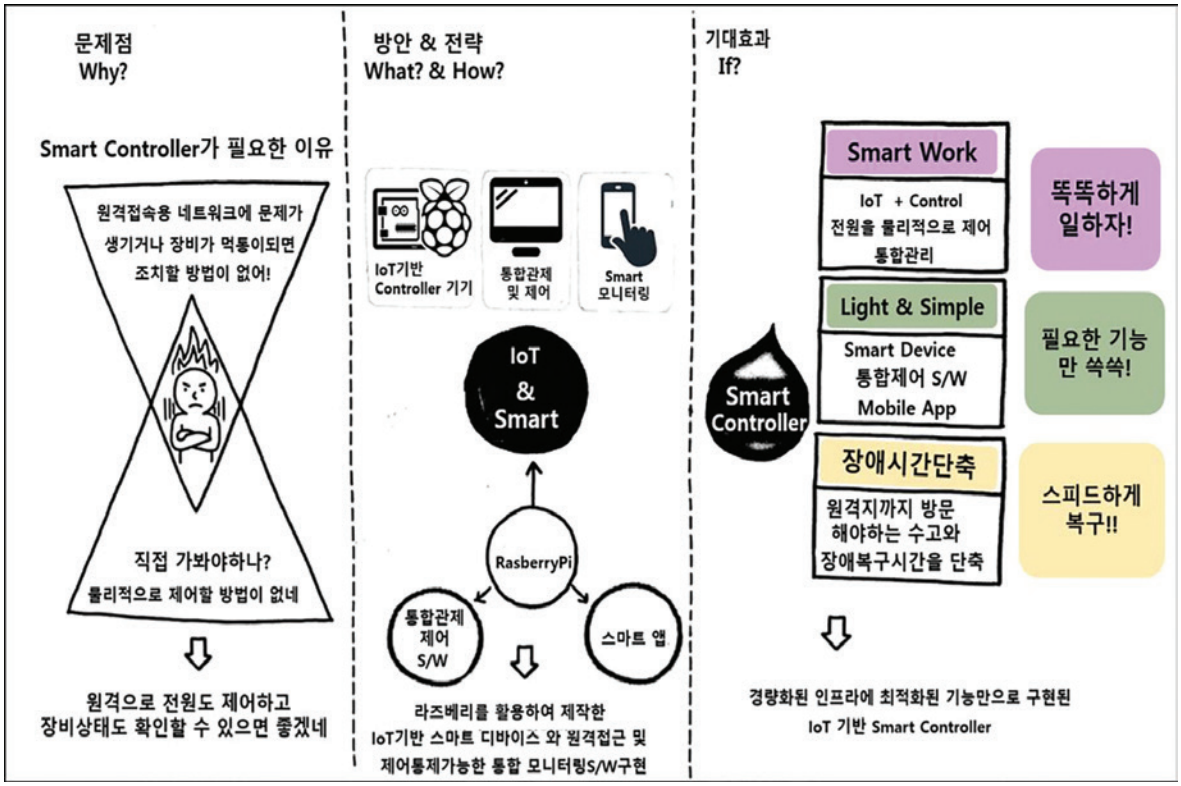

그림 2. IoT 스마트컨트롤러 제작 및 개발 배경

IoT 스마트컨트롤러 구성은?

IoT 스마트컨트롤러는 LAN과 GPIO 신호로 물리적 전원제어를 담당하는 디바이스, 제어 및 물리적 접속환경을 제공하는 통합관제 모니터링 S/W, 언제 어디서든 긴급장애 조치가 가능한 모바일 Remote 앱으로 구성된다.

※ **GPIO (General Purpose Input/Output):** 프로세서나 컨트롤러 등에서 일반 목적으로 사용하도록 준비된 입출력 포트로, 소프트웨어와 연동시키면 전기적 입력을 받거나 출력으로 특정 장치를 제어할 수 있다.

그림 3. IoT 스마트컨트롤러 구성도

IoT 스마트컨트롤러 디바이스는?

디바이스는 1RU 사이즈로 2개의 전원을 컨트롤할 수 있다. 냉각 팬을 장착해 내부 보드에서 발생하는 열을 제어하도록 구성했고, 구성 시 전원은 라즈베리파이가 5[V] 2.5[A]를 요구해 안정적으로 구동하기 위하여 전원어댑터를 3[A] 이상의 제품을 반드시 사용해야 했다. 후면에는 유지보수가 용이하도록 라즈베리파이 포트들을 외부에 배치했다.

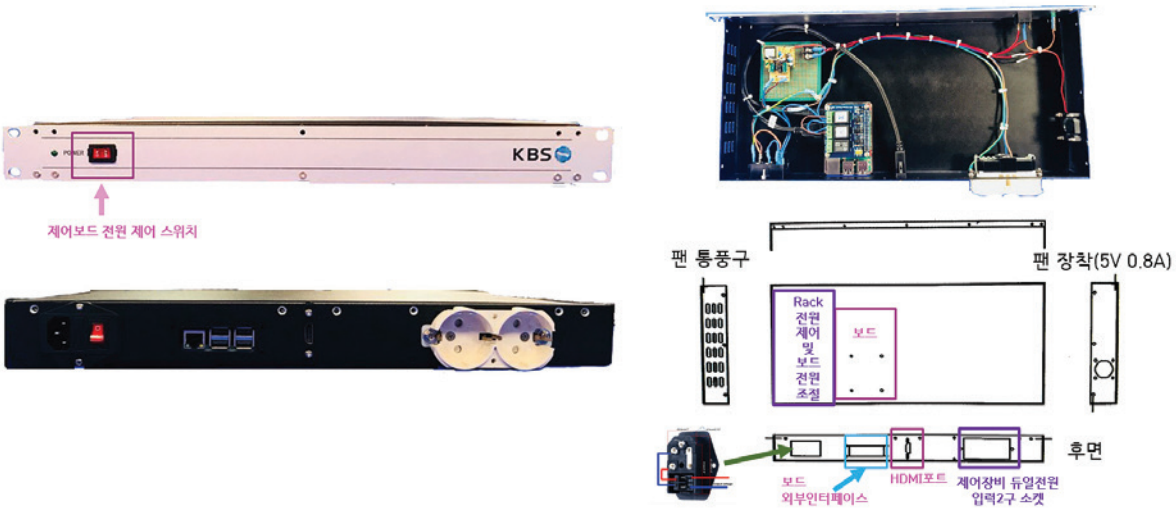

그림 4. IoT 스마트컨트롤러 외관 및 보드 구성

물리적 전원제어는?

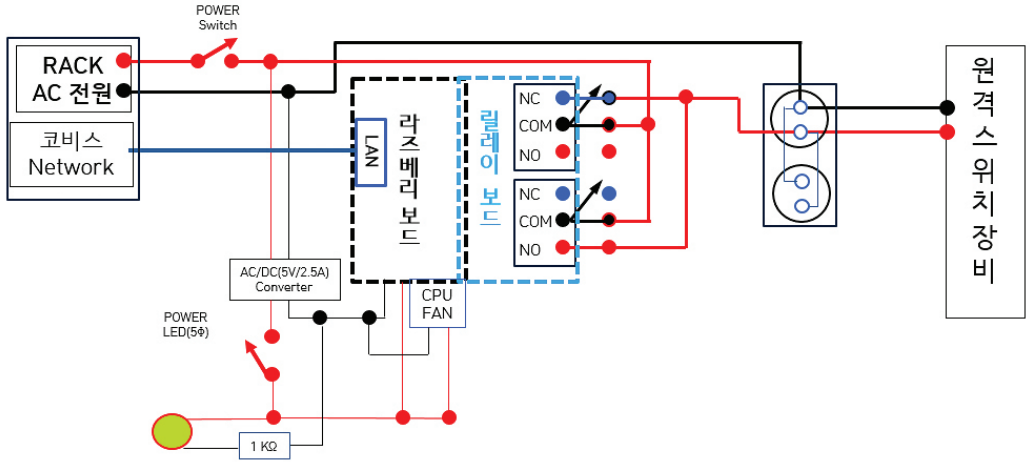

그림 5. IoT 스마트컨트롤러 디바이스 블록다이어그램

라즈베리파이 3B+로 전원제어 시스템을 구성했고, GPIO 신호를 이용해 전원선의 접점을 릴레이 모듈로 ON/OFF 하여, 물리적 전원제어를 할 수 있도록 했다. 구동은 라즈베리파이 전용 GPIO 제어기인 WebIOPi를 실행하고, 라즈베리파이의 IP 주소 포트 8000번으로 접속해 GPIO Header 메뉴에서 GPIO 37번(CH1) GPIO 38번(CH2) GPIO 39번(CH3)에서 IN/OUT을 변경하여 릴레이를 제어하도록 했다.

릴레이 모듈 채널	용도	연결 상태	동작 방법
CH1	주	NC (Normal Close)	항상 연결이 된 상태로 유지시켜 전원인가 시 잠시 연결을 끊어 전원 리셋
CH2	예비	NO (Normal Open)	항상 연결되지 않은 상태로 유지시켜 전원인가 시 잠시 연결해 전원 리셋
CH3	Hot-swap 복구용	-	

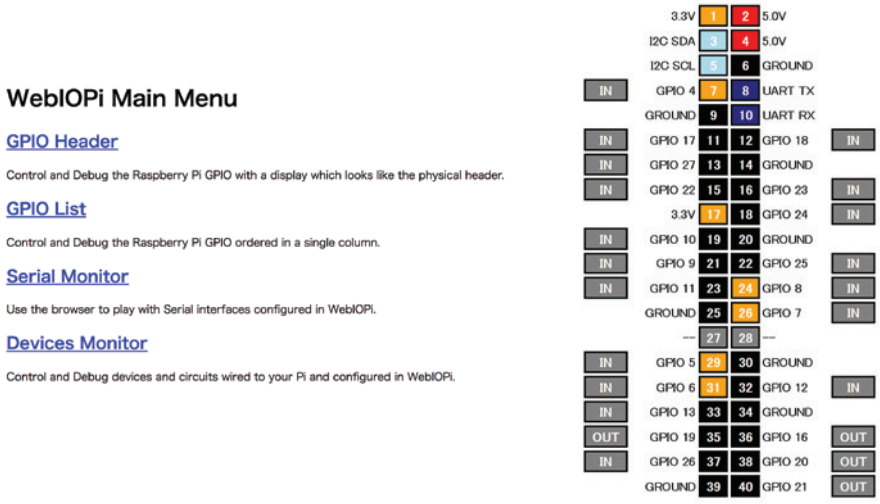

그림 6. WebIOPi GPIO 릴레이 제어화면

통합 모니터링 관제 S/W는?

소프트웨어 구성은 크게 센서부, 컨트롤부, 모니터링부로 구성되어 있고, GPIO 제어를 위해 HTTP 통신을 하도록 했다.

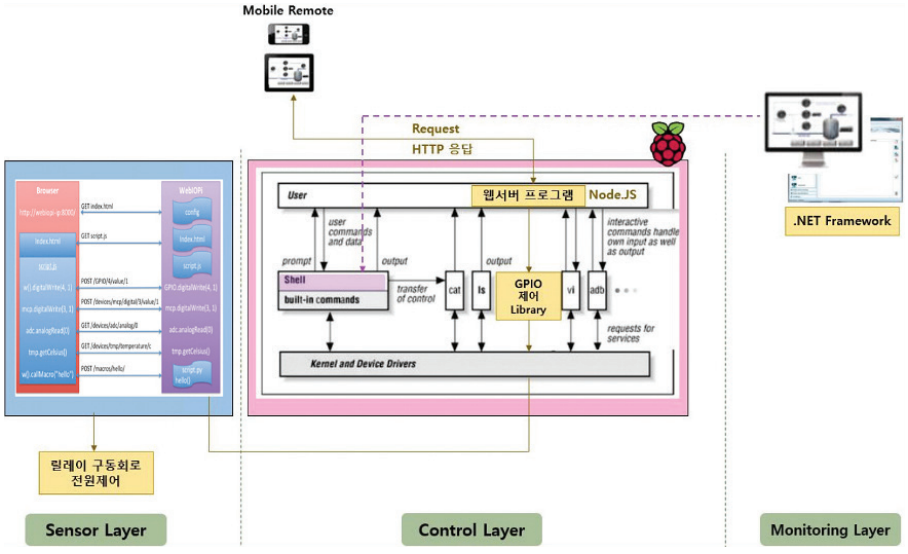

그림 7. IoT 스마트컨트롤러 소프트웨어 구성도

통합모니터링 관제 시스템은 Windows 환경에서 실행 가능한 소프트웨어로 C#과 .NET Framework를 이용해 개발된 오픈소스를 KBS 환경에 맞게 수정했다. 그리고 스크립트와 SSH 통신으로 바로 스위치에 접속해 명령어 수행 및 상태 확인을 하고, WebIOPi의 REST API를 이용해 스위치 전원을 제어하도록 구현했다.

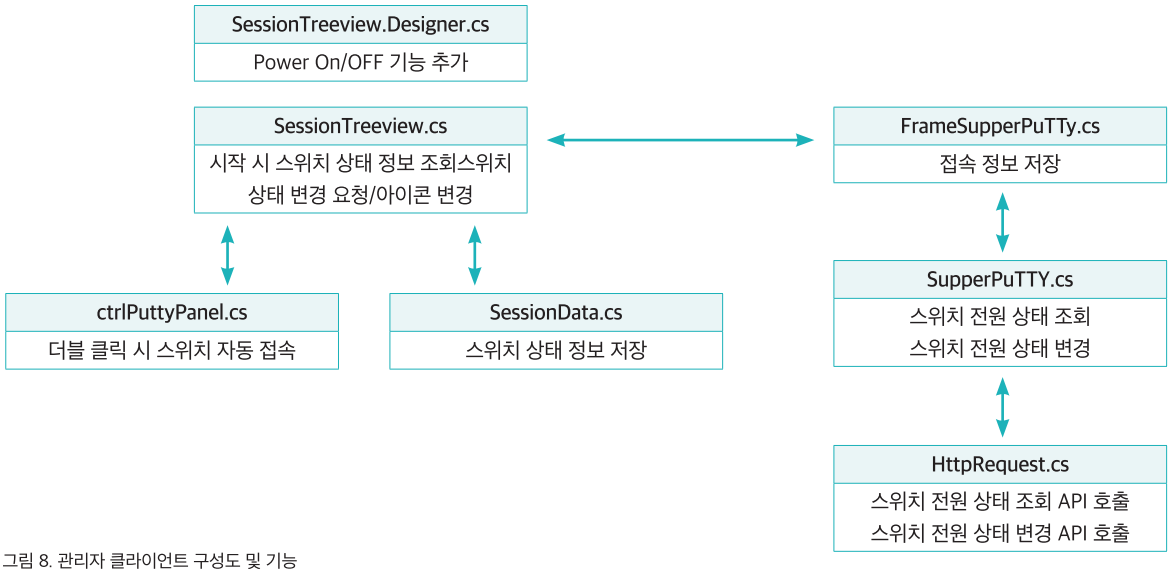

그림 8. 관리자 클라이언트 구성도 및 기능

구분	상세 기능
FrameSupperPuTTy.cs	설정 정보 및 환경을 관리하며 전체 화면 구성 관리
SupperPuTTY.cs	세션 관련 상세 기능 구현 및 스위치 전원상태 조회/변경 기능
HttpRequest.cs	HTTP 통신을 이용하여 REST API에 호출 기능
SessionTreeview.Designer.cs	세션별 메뉴 관리, Power On/OFF 메뉴 추가
SessionTreeview.cs	저장된 세션 정보를 트리 형태로 관리, 메뉴 상세 기능 구현
ctrlPuttyPanel.cs	Putty Panel 관리 및 자동 스크립트 언어 수행
SessionData.cs	세션별 정보 관리

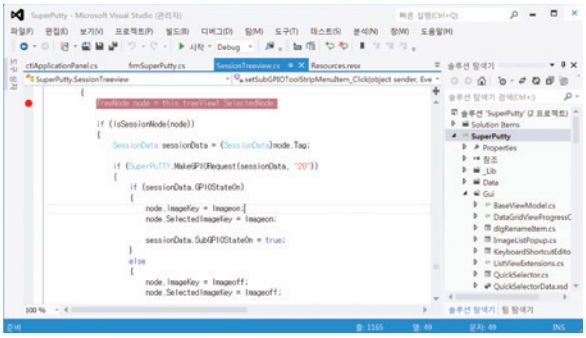

그림 9. Visual Studio 기반 개발 소스

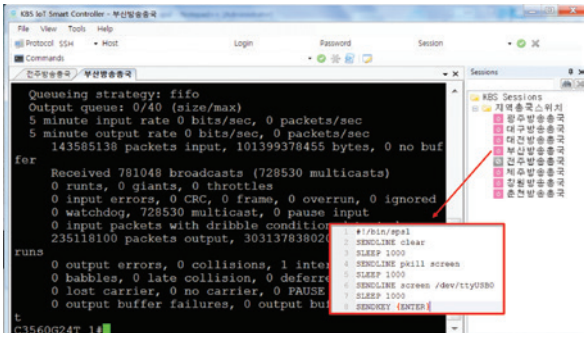

그림 10. 관리자 클라이언트를 이용하여 스위치에 접속 화면

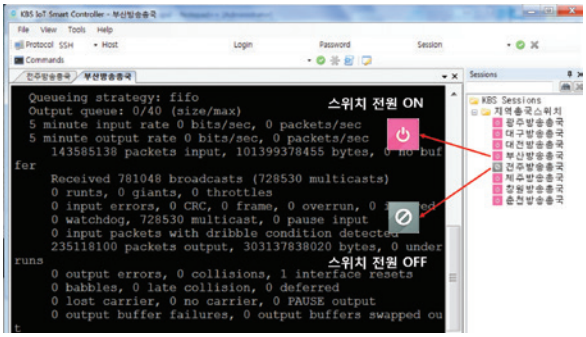

그림 11. 세션별 스위치 상태 표시

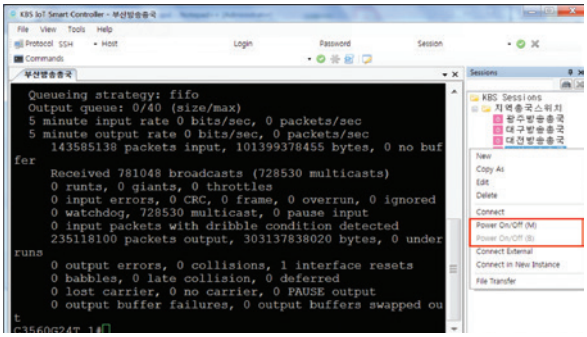

그림 12. 세션별 스위치 파워 제어 기능

모바일 제어는?

모바일 제어 소프트웨어는 전원제어 시 메인과 백업 전원의 동시 인가를 방지하고, 이벤트에 따른 결과를 나타내어 동작 여부를 확인할 수 있다.

보안을 위해 애플리케이션은 현재 사내망 Wi-Fi 환경에서만 동작하도록 개발했다.

OS	Android, IOS
개발 환경	HTML, PHP, javascript, JSON, Apache, Thunkable APP Builder

그림 13. 모바일 제어 화면

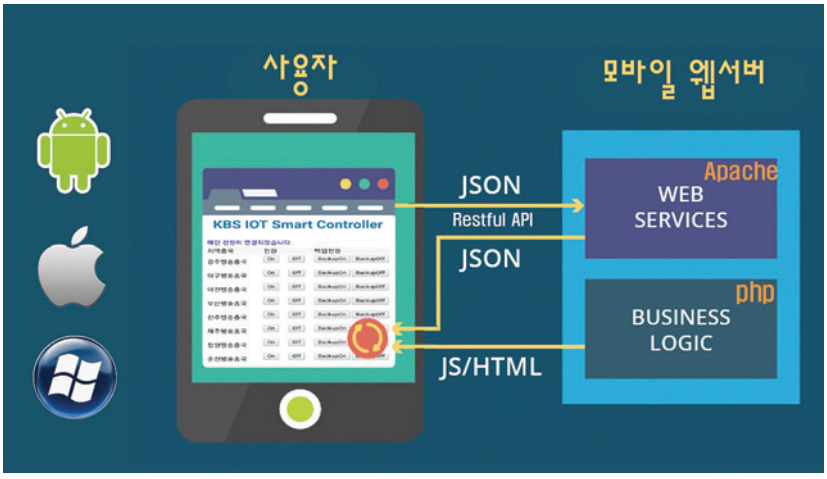

그림 14. 모바일앱 연동 아키텍처

그림 15. 연구소 백본 운용 모습

미디어창의 기술전 이후

현재 미디어창의기술전 이후에는 실제 연구소 백본 랙에 설치해 운용 중이며, 만약 상용제품으로 구성했다면 원격제어용 장비와 원격시스템제어용 장비를 별도로 구매해야 하는데, IoT 스마트컨트롤러로는 총 비용대비 80% 이상의 비용 절감 효과를 볼 수 있었다.

이번 출품작은 스위치 장비에 맞추어 진행했지만, 일반적인 방송장비 및 사무장비의 전원에도 무리 없이 사용할 수 있기 때문에 전원작업 등으로 원격지 장비의 전원상태를 확인해야 할 경우 IoT 스마트컨트롤러 사용으로 장비 전원을 효율적으로 관리할 수 있다.

출품작을 준비하면서

4개월간 준비를 하면서, 서로 다른 부서의 사람들이 모여서 원활한 커뮤니케이션으로 지식과 경험을 공유하고, 리더십을 더하면 상당히 좋은 결과물이 나온다는 것을 느낄 수 있었습니다.

창원 - 서울 간 먼 거리에도 불구하고, 많이 부족하지만 좋은 피드백들로 이끌어주신 강자원, 김정원, 이승일 선배 감사드립니다.

그림 16. 출품작 마무리 후 (왼쪽부터 배인탁 이승일 김정원 강자원)