

사물인터넷 프로토콜을 이용한 태양광 관리 시스템

김은숙*, 최지현*, 이동엽*, 양효식*
세종대*

Solar Power Management System using IoT Protocol

Eun-Sook Kim*, Jihyun Choi*, Dongyeop Lee*, Hyosik Yang*
Sejong University*

Abstract - 태양광 에너지는 유지보수가 간편하고 장기간 운용이 가능한 신재생 에너지이다. 태양광 발전이 진행될수록 생산된 에너지를 저장, 모니터링, 제어가 가능한 태양광 관리 시스템이 중요하다. 태양광 관리 시스템을 개발하기 위해서는 통신이 필수적인 기능이 된다. 한정된 자원을 가진 라즈베리파이에서 통신하기 위해서는 경량화된 프로토콜인 사물인터넷 프로토콜이 필요하다. 본 논문에서는 사물인터넷 프로토콜 중 MQTT 프로토콜을 이용하여 태양광 관리 시스템 모의 테스트 베드 개발 과정에 대하여 서술한다. 아두이노, 라즈베리파이 등을 활용하여 전력 값을 계측하고 통신을 통해 서버에서 값을 모니터링하며 서버에서 송신하는 신호로 서버를 제어한다. 이를 통해 MQTT 프로토콜을 이용한 통신으로 정보를 주고받음으로써 사물인터넷 프로토콜로 태양광 관리가 가능하다는 것을 제시한다.

1. 서 론

태양광발전은 태양의 일조량을 이용하여 얻은 에너지를 전력으로 변환한 것을 말한다. 유지보수가 간편하고 가정, 공장 등에서 장기간 운용 가능하다는 점에서 주목받는 신재생 에너지이다. 태양광 발전이 발전할수록 생산된 에너지를 저장, 모니터링, 제어가 가능한 태양광 관리 시스템의 중요성이 더욱 커지고 있다.

태양광 관리 시스템을 개발하기 위해서는 통신이 필수적인 기능이 된다. 라즈베리파이에서 전력 값을 지속해서 측정하고 서버에서는 전력 값을 모니터링하기 위해서는 통신이 필요하고 한정된 자원을 가진 라즈베리파이에서 통신하기 위해서는 경량 프로토콜인 사물인터넷 프로토콜이 필요하다. 본 논문에서는 사물인터넷 프로토콜 중 MQTT (Message Queuing Telemetry Transport)를 사용한 통신으로 태양광 관리 시스템 모의 테스트 베드의 개발 과정에 관해 서술한다 [1].

다른 태양광 관리 시스템과 비교하여 스마트폰 애플리케이션과 태양광 관리 회로가 가까운 거리에 위치하였을 때 사물인터넷 프로토콜 대신 모바일 블루투스 통신을 이용하여 모니터링 및 제어를 하는 경우를 볼 수 있다 [2]. 본 논문의 태양광 관리 시스템은 태양광 관리 회로와 서버의 거리가 멀고 서버와 회로가 같은 장소에 있지 않아도 접근하여 제어 및 모니터링이 가능하도록 사물인터넷 프로토콜인 MQTT를 사용하여 통신을 확보한다.

2. 본 론

2.1 태양광 관리 시스템의 기능

본 논문의 태양광 관리 시스템은 다음과 같은 기능을 수행한다.

- 전원제어: 서버에서 MQTT프로토콜을 이용하여 제어 신호를 라즈베리파이로 송신하여 태양광 전원의 스위치를 제어한다.
- 전원 공급원 전환: 최대 부하 시간대 여부와 정전 여부를 파악하여 각 상황이 발생 시 전원 공급원을 일반 전원에서 태양광 전원으로 전환한다.
- 전력량 모니터링: 라즈베리파이에서 측정된 전류, 전압값으로 전력을 측정하여 서버로 송신하고 서버에서는 계측

된 값을 감시한다.

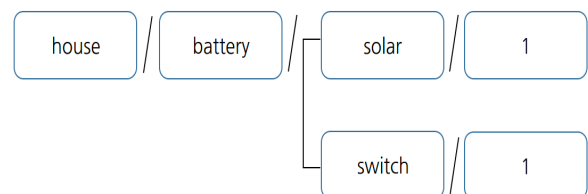
- 충전량 조절: 태양광 패널로 들어오는 전력이 배터리에 과충전 및 과방전되지 않도록 배터리의 총 충전량이 20%~80%를 유지하도록 조절한다.

2.2 태양광 관리 시스템 개발 환경

회로에서는 태양광이 일정량의 일조량을 받을 수 있도록 설정한다. 전류, 전압값 계측을 위해 전류 센서와 전압 센서를 사용하고 태양광 패널, 배터리, 태양광 컨트롤러, 부하로 구성된다. 회로를 아두이노와 연결하여 각 전류 센서와 전압 센서에서 입력되는 전류 전압값을 측정한다. 아두이노에서 측정한 값을 라즈베리파이로 전송하기 위해 시리얼 통신 환경을 설정한다.

라즈베리파이는 MQTT프로토콜을 이용한 통신을 하기 위해 paho라이브러리를 설치한다. Python으로 구성을 하며 태양광 관리 시스템에서 라즈베리파이는 Publisher와 Subscriber 역할을 한다. 서버는 MQTT 프로토콜의 broker 역할도 하므로 mosquitto 라이브러리를 설치하여 broker 역할이 가능하도록 한다. 서버는 Python으로 구성을 하며 Publisher와 Subscriber 역할을 하므로 Paho 라이브러리를 설치한다 [3][4].

MQTT는 서버와 라즈베리파이 간의 전력 값과 제어 신호를 송수신하기 위하여 사용한다. MQTT는 제한된 자원 환경에서 최적화된 IoT 프로토콜이다. 발행(Publish)/구독(Subscribe) 구조로 Publisher와 Subscriber는 Broker를 통해 통신한다. Publisher와 Subscriber가 메시지를 송수신하기 위해서는 그림 1과 같이 topic 값을 가져야 한다. Publisher가 하나의 topic과 메시지를 보내면 같은 topic을 가진 Subscriber만 송신한 메시지를 받을 수 있다. MQTT는 각 topic 별로 데이터를 전송하기에 간편하고 broadcast를 할 때도 레벨별로 토픽을 전송할 수 있다. 또한, Broker가 있어 여러 client가 통신할 때 broker에만 접근하여 자신의 토픽에 있는 메시지를 가져오면 되므로 많은 기기와의 통신에 적합하다 [5].



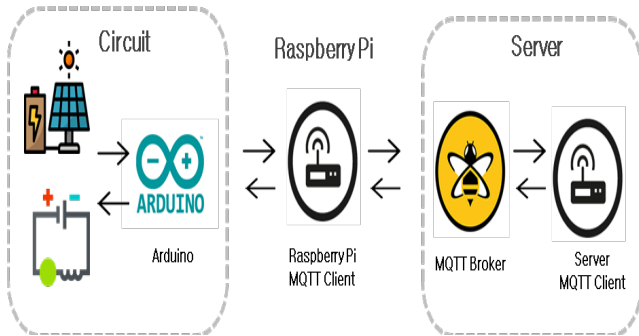
〈그림 1〉 MQTT의 topic 구조

2.3 태양광 관리 시스템의 개발 내용

태양광 관리 시스템 모의 테스트베드의 통신 아키텍처는 그림 2와 같이 크게 세 부분으로 나뉜다.

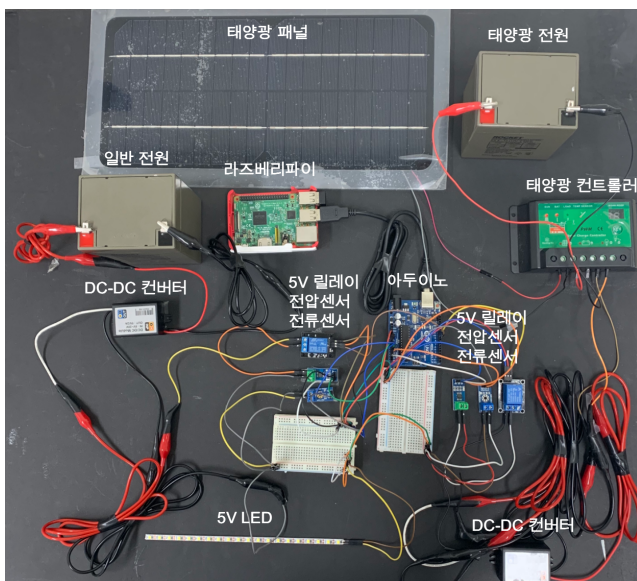
회로 부분에서는 태양광 패널로 생산된 전력을 충전하고 배터리 충전량을 조절한다. 이때 생산된 전력의 전류와 전압을 계측하여 라즈베리파이로 시리얼 통신을 한다. 라즈베리파이에서는 계측된 전류와 전압으로 총 생산된 전력량을 계산한다. 또한, 최

대부하일 때, 정전이 발생했을 때 스위치 제어를 한다. MQTT 프로토콜을 이용하여 서버로 송신하고 서버에서 전송된 제어 신호로 스위치를 제어한다. 서버에서는 전력 값을 수신하여 데이터를 저장하고 모니터링하며 태양광 배터리를 제어하는 신호를 보내는 부분으로 나뉘게 된다 [6].



〈그림 2〉 태양광 관리 시스템의 통신 아키텍처

회로 부분을 구현하기 위해 일반 전원을 대체할 배터리와 태양광에서 생산된 전력을 충전할 배터리를 설정한다. 태양광 배터리를 충전하기 위해 태양광 패널, 태양광 패널 컨트롤러, 배터리를 연결한다. 태양광 패널에서 생산된 전력이 배터리와 연결됨으로써 역전류가 흐르지 않도록 조절하고 충전량이 20%~80%를 유지하도록 태양광 컨트롤러를 구성한다. ACS712 전류 센서, DM452 전압 센서를 부하에 연결하여 Hole Effect 효과를 발생시킴으로써 전류가 흐를 수 있도록 하고 릴레이를 전류 센서와 배터리에 연결하여 제어가 가능하도록 한다. 완성된 회로를 아두이노와 연결하고 전류와 전압을 계측한다. 그림 3은 태양광 관리 시스템의 회로를 보여준다.



〈그림 3〉 태양광 관리 시스템의 회로

아두이노와 라즈베리파이로 시리얼 통신을 하여 계측한 값을 전송하고 라즈베리파이는 계측된 전류, 전압값으로 총생산 전력을 계산하여 MQTT broker 서버에 송신한다. 서버의 topic은 /home/server로 설정한다. 또한, 라즈베리파이에서는 전원 공급원을 전환한다. 최대 부하 시간대를 서버로부터 받아 저장하고 현재의 시간과 최대 부하 시간대를 비교한다. 만약 최대 부하 시간대에 해당한다면 먼저 태양광 전원의 스위치를 on으로 설정하고 일반 전원의 스위치를 off로 설정한다. 일반 전원을 먼저 끄게 되면 전원을 전환하는 과정에서 정전이 발생하므로 태양광 전원이 완전히 켜진 상태에서 일반 전원을 켜야 한다. 배터리를

병렬로 연결하여 전압의 세기를 조절하여 스위치 제어를 한다. 최대 부하 시간대가 종료되면 다시 일반 전원을 공급받기 위해 일반 전원의 스위치를 on으로 변경한 후에 태양광 전원을 off로 바꾸어 기존의 상태로 돌아가도록 한다. 정전 발생 시에는 전압 값이 떨어지면 전원 공급이 되지 않는 것으로 판단하여 태양광 전원의 스위치를 on을 한다. 일반 전원의 스위치는 off를 하지 않은 채로 유지하고 전압값이 정상적인 상태로 회복하면 태양광 전원의 스위치를 off로 변환시킨다.

서버에서는 전력 값을 수신하여 지속적으로 전력 값을 모니터링이 가능하다. 데이터를 사용자의 ID 별로 CSV 포맷 파일에 저장하고 데이터베이스에 사용자 ID 와 폴더명을 같이 저장하여 데이터를 저장 및 처리를 한다. CSV 포맷 파일에서는 많은 전력 값을 가지게 된다면 용량이 비대해지므로 1시간 단위로 압축을 하고 24시간이 지나면 하루 단위로 압축을 진행한다. 전력을 압축함으로써 파일의 용량을 줄이고 전력 사용량을 기간별로 보기 쉽게 파악할 수 있다. 서버에서는 /home/battery/solar/1 topic으로 전송된 데이터를 broker로부터 수신하고 태양광 전원 제어를 위한 신호를 /home/battery/switch/1의 topic으로 전송한다. 전력 값을 지속적으로 받기 위하여 loop를 계속 돌아가게 하고 구독하여 연속적인 값들을 모니터링할 수 있다.

3. 결 론

본 논문에서는 MQTT를 이용하여 태양광에서 생산되는 에너지 전력을 계측하고 모니터링 및 제어를 할 수 있는 태양광 관리 시스템 모의 테스트베드를 구현하였다. 아두이노와 라즈베리파이로 전력 값을 계측하고 서버와 MQTT프로토콜 통신으로 정보를 주고받음으로써 사물인터넷 프로토콜로 관리가 가능함을 파악하였다. 제한된 네트워크 환경에서도 MQTT의 Broker와 연결이 된다면 여러 클라이언트와 정보를 주고받음으로써 사물인터넷 프로토콜의 사용이 확장성이 보장된다는 것을 알 수 있다. 사물인터넷 프로토콜을 모의 테스트 베드뿐만 아니라 대단위 아파트 등에 적용하게 된다면 다른 프로토콜보다 손쉽게 태양광 관리가 가능할 것으로 판단된다.

감사의 글

이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.NRF-2016R1D1A1B03935785) 또한 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음 (2015-0- 00938).

[참 고 문 헌]

- [1]. 박용신, “전력계측 및 제어를 위한 MQTT protocol 기반 모바일 클라이언트 개발”, 중앙대학교 석사 학위논문, 2017
- [2]. 제현우, “스마트 그리드를 위한 태양광 인버터 시스템”, 청주대학교 석사 학위논문, 2013
- [3]. “https://mosquitto.org/download/”, mosquitto, 2019
- [4]. “https://projects.eclipse.org/projects/iot.paho/releases/1.4.0- photon”, Eclipse, 2018
- [5]. “MQTT Version 5.0”, OASIS (Organization for the Advancement of Structured Information Standards), 2019
- [6]. 전현지, 이동진, 양효식, “XMPP와 MQTT를 이용한 마이크로 그리드 데이터 교환 성능 분석”, 대한전기학회 하계학술대회 논문집, 2018년도