



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월02일  
(11) 등록번호 10-2285352  
(24) 등록일자 2021년07월28일

(51) 국제특허분류(Int. Cl.)  
H04L 12/24 (2006.01) H04L 29/08 (2006.01)  
H04L 29/12 (2006.01)  
(52) CPC특허분류  
H04L 41/0226 (2013.01)  
H04L 61/1552 (2013.01)  
(21) 출원번호 10-2019-0154970  
(22) 출원일자 2019년11월28일  
심사청구일자 2019년11월28일  
(65) 공개번호 10-2021-0066114  
(43) 공개일자 2021년06월07일  
(56) 선행기술조사문헌  
KR1020190057806 A\*  
KR1020180061150 A\*  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
세종대학교산학협력단  
서울특별시 광진구 능동로 209 (군자동, 세종대학교)  
(72) 발명자  
송재승  
서울특별시 강서구 강서로 348 우장산힐스테이트  
110동 902호  
손현서  
서울특별시 광진구 광나루로15길 61, 드림빌 503호  
(74) 대리인  
이준성

전체 청구항 수 : 총 13 항

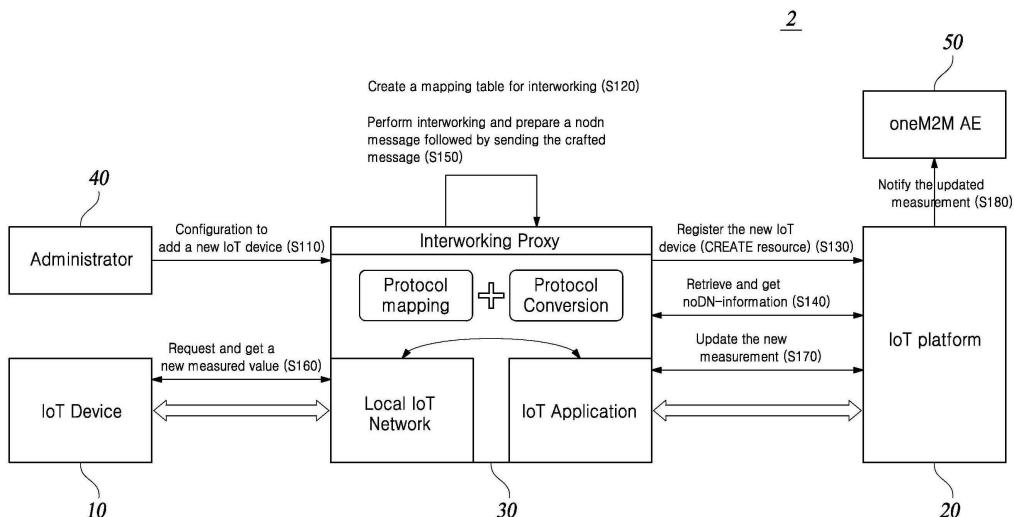
심사관 : 윤태섭

(54) 발명의 명칭 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시, 방법 및 상기 프록시를 포함하는 시스템

(57) 요약

본 발명의 실시예에 따른 이기종 IoT 장치와 IoT 플랫폼의 연동 방법은, 미등록 장치의 장치 ID, 리소스 아이디 및 프로토콜정보를 포함하는 맵핑테이블을 생성하는 단계; 상기 맵핑테이블에 기초하여, 상기 장치의 리소스를 생성하는 단계; 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하는 단계; 및 상기 커맨드에 대응하여 상기 장치로부터 장치정보가 수신되면, 상기 장치정보를 갖도록 상기 리소스를 업데이트하는 단계를 포함한다.

대표도



(52) CPC특허분류

- H04L 67/1074* (2013.01)
- H04L 67/12* (2013.01)
- H04L 67/16* (2013.01)
- H04L 67/2842* (2013.01)
- H04L 69/08* (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711096548
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	미래선도기술개발
연구과제명	솔라 사이너지용 반투명 태양전지 및 사물인터넷 기술 개발
기 여 율	1/1
과제수행기관명	중앙대학교 산학협력단
연구기간	2018.12.03 ~ 2022.04.02
공지예외적용	: 있음

---

## 명세서

### 청구범위

#### 청구항 1

프록시를 이용하여, 서로 상이한 프로토콜을 사용하는 이기종 IoT 장치와 IoT 플랫폼을 연동하는 방법으로서, 상기 IoT 플랫폼은 리소스 기반의 구조(RoA; Resource Oriented Architecture)를 갖고 IoT 장치로부터 수신된 장치 정보를 리소스에 저장 및 관리하고,

상기 방법은,

프록시가, 미등록 IoT 장치의 장치 ID, 리소스 아이디 및 프로토콜정보를 포함하는 맵핑테이블을 생성하는 단계;

프록시가, 상기 맵핑테이블에 기초하여, 상기 장치의 리소스-상기 리소스 아이디 및 상기 프로토콜정보를 포함-를 생성하여 IoT 플랫폼에 등록하는 단계;

프록시가, 상기 IoT 플랫폼에 상기 장치의 프로토콜정보를 요청하여 상기 IoT 플랫폼으로부터 상기 리소스에 저장된 상기 장치의 프로토콜정보를 수신하는 단계;

프록시가, 수신된 프로토콜에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 상기 장치에 전송하는 단계; 및

프록시가, 상기 커맨드에 대응하여 상기 장치로부터 장치정보가 수신되면, 상기 장치정보를 갖도록 상기 리소스를 업데이트하는 단계;

를 포함하고,

상기 리소스에 저장되는 프로토콜정보는, 상기 장치의 프로토콜의 종류 및 상기 장치의 프로토콜의 레지스터 파일 정보를 포함하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 2

제1항에 있어서,

상기 IoT 플랫폼이, 상기 리소스가 업데이트된 후, 상기 리소스를 구독하는 애플리케이션 엔티티에 상기 장치정보를 통지하는 단계

를 더 포함하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 3

제1항에 있어서,

상기 IoT 플랫폼은 oneM2M 플랫폼인 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 4

제3항에 있어서,

상기 리소스는 flexContainer인 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 5

제3항에 있어서,

상기 장치정보는 상기 리소스의 자식 리소스(child resource) 또는 속성(attribute)으로서 저장되는 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 6

삭제

#### 청구항 7

제1항에 있어서,

상기 장치의 프로토콜은 Modbus이고,

상기 리소스는 Modbus의 레지스터 파일 정보를 포함하는 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동 방법.

#### 청구항 8

서로 상이한 프로토콜을 사용하는 이기종 IoT 장치와 IoT 플랫폼을 연동하기 위한 프록시로서,

상기 IoT 플랫폼은 리소스 기반의 구조(RoA; Resource Oriented Architecture)를 갖고 IoT 장치로부터 수신된 장치정보를 리소스에 저장 및 관리하고,

상기 프록시는,

미등록 IoT 장치의 장치 ID, 리소스 아이디 및 프로토콜 정보를 포함하는 맵핑테이블이 저장된 맵핑테이블 저장부;

상기 맵핑테이블에 기초하여, IoT 플랫폼에 상기 장치의 리소스-상기 리소스 아이디 및 상기 프로토콜정보를 포함함-를 생성하여 등록하는 리소스 관리부; 및

상기 IoT 플랫폼에 상기 장치의 프로토콜정보를 요청하여 상기 IoT 플랫폼으로부터 상기 리소스에 저장된 상기 장치의 프로토콜정보를 수신하고, 수신된 프로토콜정보에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 상기 장치에 전송하고, 상기 커맨드에 대응하여 상기 장치로부터 장치정보를 수신하는 장치 통신부;

를 포함하고,

상기 리소스 관리부는, 상기 장치 통신부에서 수신된 장치정보를 이용하여 상기 리소스를 업데이트하고,

상기 리소스에 저장되는 프로토콜정보는, 상기 장치의 프로토콜의 종류 및 상기 장치의 프로토콜의 레지스터 파일 정보를 포함하는, 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시.

#### 청구항 9

제8항에 있어서,

상기 IoT 플랫폼은 oneM2M 플랫폼인 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시.

#### 청구항 10

제9항에 있어서,

상기 리소스는 flexContainer인 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시.

**청구항 11**

제10항에 있어서,

상기 장치정보는 상기 리소스의 자식 리소스(child resource) 또는 속성(attribute)으로서 저장되는 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시.

**청구항 12**

삭제

**청구항 13**

제8항에 있어서,

상기 장치의 프로토콜은 Modbus이고,

상기 리소스는 Modbus의 레지스터 파일 정보를 포함하는 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시.

**청구항 14**

미등록 IoT 장치;

상기 장치와 상이한 프로토콜을 사용하고, 리소스 기반의 구조(RoA; Resource Oriented Architecture)를 갖고 IoT 장치로부터 수신된 장치 정보를 리소스에 저장 및 관리하는 IoT 플랫폼; 및

상기 장치의 장치 ID, 리소스 아이디 및 프로토콜 정보를 포함하는 맵핑테이블이 저장된 맵핑테이블 저장부와, 상기 맵핑테이블에 기초하여, 상기 IoT 플랫폼에 상기 장치의 리소스-상기 리소스 아이디 및 상기 프로토콜정보를 포함함-를 생성하여 등록하는 리소스 관리부와, 상기 IoT 플랫폼에 상기 장치의 프로토콜정보를 요청하여 상기 IoT 플랫폼으로부터 상기 리소스에 저장된 상기 장치의 프로토콜정보를 수신하고, 수신된 프로토콜정보에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 상기 장치에 전송하고, 상기 커맨드에 대응하여 상기 장치로부터 장치정보를 수신하는 장치 통신부를 포함하고, 상기 리소스 관리부는, 상기 장치 통신부에서 수신된 장치정보를 이용하여 상기 리소스를 업데이트하고, 상기 리소스에 저장되는 프로토콜정보는, 상기 장치의 프로토콜의 종류 및 상기 장치의 프로토콜의 레지스터 파일 정보를 포함하는, 프록시;

를 포함하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 시스템.

**청구항 15**

제14항에 있어서,

상기 IoT 플랫폼은,

상기 리소스가 상기 장치정보로 업데이트된 후, 상기 리소스를 구독하는 애플리케이션 엔티티에 상기 업데이트된 장치정보를 통지하는 것을 특징으로 하는 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 시스템.

**발명의 설명**

**기술 분야**

이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시 및 상기 프록시를 포함하는 시스템과 이기종 IoT 장치와

[0001]

IoT 플랫폼의 연동 방법에 관한 것이다.

**배경 기술**

- [0003] 인터넷은 인간이 정보를 생성하고 소비하는 인간 중심의 연결 망에서, 사물 등 분산된 구성 요소들 간에 정보를 주고 받아 처리하는 IoT(internet of things, 사물인터넷) 망으로 진화하고 있다. 또한, IoT를 구현하기 위해 센싱 기술, 유무선 통신 및 네트워크 인프라, 서비스 인터페이스 기술, 및 보안 기술과 같은 기술 요소 들이 요구되어, 최근에는 사물간의 연결을 위한 센서 네트워크(sensor network), 사물 통신(machine to machine, M2M), MTC(machine type communication)등의 기술이 연구되고 있다.
- [0004] IoT 특성상 다양한 네트워크 프로토콜을 사용하는 IoT 장치가 다수 존재한다. 이기종 IoT 장치들은 IoT 플랫폼을 통해 연결되고, IoT 플랫폼을 통해 IoT 서비스에게 데이터를 전송할 수 있게 된다.
- [0005] 하지만, 기존의 이기종 IoT 장치와 IoT 플랫폼의 연결은 개발자의 독자적인 구현으로 이루어지고 있다. 특히, IoT 플랫폼이 지원하지 않는 프로토콜을 사용하는 IoT 장치에서 IoT 플랫폼으로 메시지를 전달하기 위해서는 필요한 모든 정보들이 인터워킹을 제공하는 프록시의 내부 메모리에 저장되어 관리되어 왔다.
- [0006] 이 경우 해당 프록시의 재사용이 불가능하며, 상호연동을 위해 프록시에 과도한 프로토콜 매핑 정보들이 지속적으로 저장되어야 해서 메모리 낭비가 발생하였다.
- [0007] IoT 국제 표준인 oneM2M 표준 문서에 따르면, 기존의 IoT 아키텍처는 모든 사물을 리소스화 하고 이를 유니크한 URI를 통해서 접근이 가능하도록 정의한다. 하지만, 이러한 리소스에 이기종 프로토콜의 매핑 정보를 저장하기 위한 내용이 정의되어 있지 않아, 기존 프록시는 모든 정보들을 내부 메모리에 저장하여 상호연동을 진행해야 했다.

**선행기술문헌**

**특허문헌**

- [0009] (특허문헌 0001) 미국공개특허공보 제2018/0063879호

**발명의 내용**

**해결하려는 과제**

- [0010] 본 발명은 이기종 IoT 장치와 표준 기반 IoT 플랫폼간의 상호연동을 위한 프록시와, 프록시를 포함하는 시스템 및, 이기종 IoT 장치와 IoT 플랫폼의 연동 방법을 제공하고자 한다.

**과제의 해결 수단**

- [0012] 본 발명의 실시예에 따른 이기종 IoT 장치와 IoT 플랫폼의 연동 방법은, 미등록 장치의 장치 ID, 리소스 아이디 및 프로토콜정보를 포함하는 맵핑테이블을 생성하는 단계; 상기 맵핑테이블에 기초하여, 상기 장치의 리소스를 생성하는 단계; 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하는 단계; 및 상기 커맨드에 대응하여 상기 장치로부터 장치정보가 수신되면, 상기 장치정보를 갖도록 상기 리소스를 업데이트하는 단계를 포함한다.
- [0013] 상기 리소스가 업데이트된 후, 상기 리소스를 구독하는 애플리케이션 엔티티에 상기 장치정보를 통지하는 단계를 더 포함할 수 있다.
- [0014] 상기 IoT 플랫폼은 oneM2M 플랫폼일 수 있다.
- [0015] 상기 리소스는 flexContainer일 수 있다.
- [0016] 상기 장치정보는 상기 리소스의 자식 리소스(child resource) 또는 속성(attribute)으로서 저장될 수 있다.
- [0017] 상기 리소스는 상기 프로토콜정보를 포함할 수 있다.
- [0018] 상기 장치의 프로토콜은 Modbus이고, 상기 리소스는 Modbus의 레지스터 파일 정보를 포함할 수 있다.

[0019] 본 발명의 실시예에 따른 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 프록시 장치는, 미등록 장치의 장치 ID, 리소스 아이디 및 프로토콜 정보를 포함하는 맵핑테이블이 저장된 맵핑테이블 저장부; 상기 맵핑테이블에 기초하여, IoT 플랫폼에 상기 장치의 리소스를 생성하는 리소스 관리부; 및 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하고, 상기 커맨드에 대응하여 상기 장치로부터 장치정보를 수신하는 장치 통신부를 포함하고, 상기 리소스 관리부는, 상기 장치 통신부에서 수신된 장치정보를 이용하여 상기 리소스를 업데이트한다.

[0020] 이기종 IoT 장치와 IoT 플랫폼의 연동을 위한 시스템은, 미등록 장치; IoT 플랫폼; 및 상기 장치의 장치 ID, 리소스 아이디 및 프로토콜 정보를 포함하는 맵핑테이블이 저장된 맵핑테이블 저장부와, 상기 맵핑테이블에 기초하여, 상기 IoT 플랫폼에 상기 장치의 리소스를 생성하는 리소스 관리부와, 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하고, 상기 커맨드에 대응하여 상기 미등록 장치로부터 장치정보를 수신하는 장치 통신부를 포함하고, 상기 리소스 관리부는, 상기 장치 통신부에서 수신된 장치정보를 이용하여 상기 리소스를 업데이트하는, 프록시 장치를 포함한다.

[0021] 상기 IoT 플랫폼은, 상기 리소스가 상기 장치정보로 업데이트된 후, 상기 리소스를 구독하는 애플리케이션 엔티티에 상기 업데이트된 장치정보를 통지할 수 있다.

**발명의 효과**

[0023] 본 발명의 실시예에 의하면, 표준적인 방법으로 이기종 IoT 장치와 표준 기반 IoT 플랫폼간의 상호연동이 가능하다.

[0024] 본 발명의 실시예에 의하면, 이기종 IoT 장치의 프로토콜 맵핑 정보를 저장할 수 있는 oneM2M 기반의 리소스가 제공된다. 이에 따라, 프로토콜 맵핑 정보를 프록시에 전부 저장할 필요가 없어 프록시의 메모리 부담이 경감될 수 있다.

**도면의 간단한 설명**

- [0026] 도 1은 oneM2M의 네트워크 구성을 나타낸 도면이다.
- 도 2는 본 발명의 실시예에 따른 사물인터넷 네트워크에서 프록시의 기능을 설명하기 위한 도면이다.
- 도 3은 본 발명의 실시예에 따른 사물인터넷 네트워크에서 IoT 장치와 IoT 플랫폼의 연동 방법을 설명하기 위한 도면이다.
- 도 4는 도 3의 맵핑테이블의 일 예를 나타내는 도면이다.
- 도 5는 도 3의 IoT 장치의 리소스의 일 예를 나타내는 도면이다.
- 도 6은 Modbus 레지스터 파일을 나타내는 도면이다.
- 도 7은 본 발명의 실시예에 따른 리소스를 나타내는 도면이다.
- 도 8은 본 발명의 실시예에 따른 리소스와 맵핑테이블의 대응을 설명하기 위한 도면이다.
- 도 9는 도 7의 리소스를 이용하여 생성된, 배터리의 레벨을 읽어내기 위한 커맨드를 나타내는 도면이다.
- 도 10은 본 발명의 실시예에 따른 사물인터넷 네트워크의 구성을 나타내는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0027] 본 명세서에서 설명된 임의의 실시예의 방법 또는 구성은 임의의 다른 방법 또는 구성에 대하여 구현될 수 있다.

[0028] 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정되어 해석되지 말아야 하며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야 한다.

[0029] 명세서 및 청구범위에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있다는 것을 의미한다.

[0030] 명세서 및 청구범위에서 용어 "포함하는"과 함께 사용될 때 단수 단어의 사용은 "하나"의 의미일 수도 있고, 또

는 "하나 이상", "적어도 하나", 및 "하나 또는 하나보다 많은"의 의미일 수도 있다.

- [0031] 명세서 및 청구범위에 기재된 "...부", "...기", "모듈", "장치" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.
- [0032] 동일한 도면 부호는 전체에 걸쳐 동일한 구성 요소를 나타낸다.
- [0033] 이하, 도면을 참조하여 본 발명의 실시예에 대하여 구체적으로 설명한다.
- [0035] 도 1은 oneM2M의 네트워크 구성을 나타낸 도면이다.
- [0036] 도 1을 참조하면, 상기 oneM2M의 네트워크는 애플리케이션 전용 노드 (ADN: Application Dedicated Node)(101), 애플리케이션 서비스 노드(ASN: Application Service Node)(103), 중간노드(MN: Middle Node)(105) 및 인프라스트럭처 노드(IN: Infrastructure Node)(107)로 구성될 수 있다.
- [0037] 애플리케이션 전용 노드(101)는 사물인터넷 애플리케이션을 포함하는 장치로써, 사물인터넷 서비스 로직만을 포함하여 제한된 기능을 가지는 제한적인 (Constrained) 장치를 의미한다. 애플리케이션 서비스 노드(103)는 사물인터넷 애플리케이션뿐만 아니라 공통 서비스를 제공하는 사물인터넷 장치를 의미한다. 중간 노드(105)는 다른 노드들과 인프라스트럭처 노드(107)를 연결해주는 사물인터넷 게이트웨이이며, 인프라스트럭처 노드(107)는 네트워크 인프라스트럭처에 위치해 사물인터넷 서비스를 제공하는 사물인터넷 서버가 될 수 있다.
- [0038] 상기 각 노드는 하나 이상의 애플리케이션 엔터티 (AE: Application Entity) 또는 공통 서비스 엔터티 (CSE: Common Services Entity)를 포함할 수 있다.
- [0039] 상기 애플리케이션 엔터티는 단대단 (end-to-end) 사물인터넷 솔루션을 위한 애플리케이션 로직을 제공한다.
- [0040] 상기 공통 서비스 엔터티는 사물인터넷의 다양한 애플리케이션 엔터티들이 공통적으로 사용 가능한 공통 서비스 기능들을 제공한다. 상기 공통 서비스 기능에는 데이터 관리 및 저장 기능, 통신 관리 및 전달 처리 기능, 등록 (registration) 기능, 보안 기능, 구독 및 통지 기능(Subscription/ Notification), 그룹 관리 기능 등이 있다.
- [0041] 상기 구독 및 통지 기능은 애플리케이션 엔터티 또는 공통 서비스 엔터티가 어떤 리소스에 대한 특정 조건의 변경을 구독(Subscription)하면 해당 리소스가 변경되어 조건 만족 시 이를 통지(Notification)하는 역할을 수행한다.
- [0042] oneM2M은 리소스 기반의 구조(RoA: Resource Oriented Architecture)를 따른다. 공통 서비스 기능은 리소스에 대한 동작(Operation)을 통해서 제공된다. 리소스는 트리 구조(Tree Structure)를 가지며, 고유한 주소(예, URI)를 이용하여 어드레싱(addressing) 될 수 있다. 상기 리소스는 공통 서비스 엔터티에 저장되며 애플리케이션 엔터티는 리소스를 가질 수 없다.
- [0043] 상기 리소스는 다양한 리소스 타입들을 포함하며, 각각의 리소스 타입은 속성들(Attributes)과 자식 리소스 (Child Resource) 타입들로 정의된다. 예를 들어, 애플리케이션 엔터티 리소스 타입은 애플리케이션 엔터티와 관련된 정보(예, 애플리케이션 이름, 네트워크 주소, 시멘틱 정보)들의 속성들을 포함하며 애플리케이션 엔터티에서 발생하는 정보를 저장하기 위해 컨테이너 리소스 타입, 그룹에 동작 전달을 위해 그룹 리소스 타입을 자식 리소스로 가질 수 있다. 이러한 리소스 타입은 생성 동작을 통해 실제 리소스에 생성되고, 리소스에 수정, 획득, 삭제, 통지의 동작을 통해 서비스를 사용할 수 있다.
- [0044] oneM2M 네트워크는 미등록 노드(NoDN: Non-oneM2M Device Node)(109)를 추가적으로 포함할 수 있다. 미등록 노드는 애플리케이션 엔터티나 공통 서비스 엔터티와 같은 oneM2M 엔터티를 가지지 않는 노드로서 관리 등을 포함한 상호 연동 목적으로 oneM2M 시스템에 붙어있는 노드를 나타낸다. 미등록 노드(109)는 oneM2M이 아닌 다른 프로토콜을 이용하여 통신할 수 있다.
- [0046] 도 2는 본 발명의 실시예에 따른 사물인터넷 네트워크(1)에서 프록시(30)의 기능을 설명하기 위한 도면이다.
- [0047] 도 2를 참조하면, 사물인터넷 네트워크(1)는 IoT 장치(10), IoT 플랫폼(20) 및 프록시(30)를 포함한다.
- [0048] IoT 장치(10)는 센서를 포함할 수 있으며, 센서에 의해 획득된 데이터를 특정 프로토콜에 의해 전송한다. IoT 장치(10)는 도 1의 미등록 노드(109)에 대응할 수 있다.
- [0049] IoT 플랫폼(20)은 IoT 장치(10)로부터 수신된 데이터를 이용하여 사물인터넷 서비스를 제공하는 사물인터넷



서버가 될 수 있다. IoT 플랫폼(20)은 도 1의 인프라스트럭처 노드(107)에 대응할 수 있다.

- [0050] 프록시(30)는 서로 상이한 프로토콜을 사용하는 IoT 장치(10)와 IoT 플랫폼(20)의 프로토콜 맵핑 및 프로토콜 변환을 통해 IoT 장치(10)와 IoT 플랫폼(20)을 상호연동하기 위한 장치이다. 프록시(30)는 도 1의 중간 노드(105)에 해당할 수 있다. 예를 들어, IoT 장치(10)의 프로토콜이 Modbus이고, IoT 플랫폼(20)이 oneM2M 플랫폼인 경우, 프록시(30)는 IoT 장치(10)와의 사이에서 데이터를 송수신하는 경우에는 Modbus 프로토콜을 이용하고 IoT 플랫폼(20)과의 사이에서는 oneM2M 프로토콜을 사용함으로써, IoT 장치(10)와 IoT 플랫폼(20)을 상호연동한다.
- [0052] 본 발명의 실시예에 따른 이기종 IoT 장치와 IoT 플랫폼의 연동 방법은,
- [0053] 미등록 장치의 장치 ID, 리소스 아이디 및 프로토콜정보를 포함하는 맵핑테이블을 생성하는 단계;
- [0054] 상기 맵핑테이블에 기초하여, 상기 장치의 리소스를 생성하는 단계;
- [0055] 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하는 단계; 및
- [0056] 상기 커맨드에 대응하여 상기 장치로부터 장치정보가 수신되면, 상기 장치정보를 갖도록 상기 리소스를 업데이트하는 단계를 포함한다.
- [0058] 이하, 도 3을 참조하여 상기 이기종 IoT 장치와 IoT 플랫폼의 연동 방법의 구체적인 실시예에 대하여 설명한다.
- [0059] 도 3은 본 발명의 실시예에 따른 사물인터넷 네트워크(2)에서 IoT 장치(10)와 IoT 플랫폼(20)의 연동 방법을 설명하기 위한 도면이다.
- [0060] 도 3을 참조하면, 사물인터넷 네트워크(2)는 IoT 장치(10), IoT 플랫폼(20), 프록시(30), 관리자 장치(40) 및 애플리케이션 장치(50)를 포함할 수 있다. IoT 장치(10), IoT 플랫폼(20) 및 프록시(30)는 도 2의 사물인터넷 네트워크(1)의 IoT 장치(10), IoT 플랫폼(20) 및 프록시(30)와 동일하므로 설명을 생략한다.
- [0061] 관리자 장치(40)는 관리자가 IoT 플랫폼(20)에 접속하여 IoT 플랫폼(20)의 기능을 제어할 수 있도록 하기 위한 장치로서, 도 1의 애플리케이션 전용 노드(101)에 해당할 수 있다.
- [0062] 애플리케이션 장치(50)는 IoT 장치(10)로부터 수신된 데이터를 이용하여 특정 서비스를 제공하는 장치로서, 애플리케이션 엔터티를 포함하는 도 1의 애플리케이션 전용 노드(101)에 해당할 수 있다. 예를 들어, 애플리케이션 장치(50)는 복수의 IoT 장치(10)로부터 수신된 데이터를 수집, 분석하여 통계를 제공하거나, 통계를 기초로 전자 기기를 제어하는 기능을 수행할 수 있다.
- [0063] 도 3을 참조하면, IoT 장치(10)와 IoT 플랫폼(20)의 연동 방법은 미등록 IoT 장치(10)를 추가하는 단계(S110), 맵핑 테이블을 생성하는 단계(S120), 미등록 IoT 장치(10)의 리소스를 생성하는 단계(S130), IoT 장치(10)의 리소스로부터 IoT 장치(10)의 프로토콜정보를 수신하는 단계(S140), IoT 장치(10)로부터 장치정보를 읽어내기 위한 커맨드를 생성하는 단계(S150), IoT 장치(10)에 커맨드를 전송하고 장치정보를 수신하는 단계(S160), 수신한 장치정보를 포함하도록 IoT 장치(10)의 리소스를 업데이트하는 단계(S170), 애플리케이션 엔터티에 리소스의 변경을 통지하는 단계(S180)를 포함할 수 있다.
- [0064] 먼저, 관리자 장치(40)에 의해 미등록 IoT 장치(10)가 프록시(30)에 추가된다(S110). 관리자 장치(40)에 의해 미등록 IoT 장치(10)에 관한 정보가 프록시(30)에 입력될 수 있다.
- [0065] 다음으로, 프록시(30)는 추가된 미등록 IoT 장치(10)의 정보를 기초로 맵핑테이블을 생성한다(S120). 맵핑테이블이 이미 생성되어 있을 경우에는 미등록 IoT 장치(10)에 관한 정보를 추가하고, 생성된 맵핑테이블이 없는 경우에는 맵핑테이블을 새로 생성한다. 맵핑테이블은 IoT 장치(10)의 아이디, IoT 장치의 리소스 아이디, IoT 장치의 프로토콜 정보를 포함할 수 있다.
- [0066] 다음으로, 프록시(30)는 맵핑테이블에 기초하여 미등록 IoT 장치(10)의 리소스를 생성한다(S130). 이에 따라, IoT 장치(10)는 IoT 플랫폼(20)에 등록된다. 생성된 IoT 장치(10)의 리소스는 IoT 플랫폼(20)에 의해 관리된다. 리소스는 맵핑테이블에 저장된 리소스 아이디를 가지며, 맵핑테이블에 저장된 프로토콜 정보의 적어도 일부를 포함할 수 있다.
- [0067] 다음으로, 프록시(30)는 IoT 플랫폼(20)을 통해 IoT 장치(10)의 리소스에 저장된 프로토콜정보를 수신한다(S140).
- [0068] 다음으로, 프록시(30)는 S140 단계에서 수신된 리소스의 프로토콜정보를 이용하여, IoT 장치(10)로부터 장치정

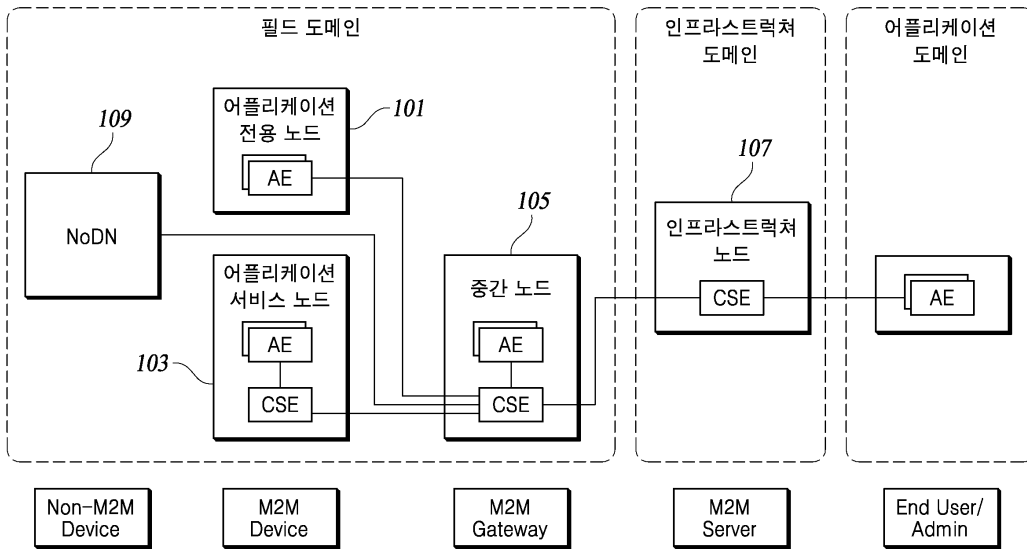
보를 읽어내기 위한 커맨드를 생성한다(S150). 여기서, 장치정보는 IoT 장치(10)가 센싱한 데이터일 수 있다. 이때, S120 단계에서 생성된 맵핑테이블의 정보도 함께 이용될 수 있다.

- [0069] 다음으로, 프록시(30)는 S150 단계에서 생성된 커맨드를 IoT 장치(10)에 전송하여, IoT 장치(10)로부터 장치정보를 수신한다(S160).
- [0070] 다음으로, 프록시(30)는 S160 단계에서 수신한 장치정보를 이용하여 IoT 장치(10)의 리소스를 업데이트한다. 이때, 프록시(30)는 IoT 플랫폼(20)을 통해 IoT 장치(10)의 리소스를 업데이트할 수 있다(S170).
- [0071] IoT 장치(10)의 리소스가 업데이트되면, IoT 플랫폼(20)은 IoT 장치(10)의 리소스를 구독하는 애플리케이션 엔터티에 업데이트된 장치정보를 통지한다(S180).
- [0073] 도 4는 도 3의 맵핑테이블의 일 예를 나타내는 도면이다.
- [0074] 도 4를 참조하면, 맵핑테이블은, IoT 장치의 아이디, IoT 장치의 리소스 아이디, IoT 장치의 프로토콜정보를 포함하며, 추가적으로 생성시간 및 사용가능성이 포함될 수 있다.
- [0075] 프록시(30)는 관리자 장치(40)로부터 IoT 장치(10)의 정보가 입력되면, 이를 이용하여 맵핑테이블을 생성한다. IoT 장치의 리소스 아이디는 추후 생성될 IoT 장치의 리소스를 가리킨다. 프로토콜정보는 IoT 장치가 사용하는 프로토콜명뿐만 아니라 프로토콜과 관련된 정보를 포함할 수 있다. 프로토콜정보는 추후 리소스에 포함되는 것으로, IoT 장치(10)와 IoT 플랫폼(20)이 연동하기 위해 필요한 프로토콜에 관한 정보를 의미한다.
- [0076]
- [0077] 도 5는 도 3의 IoT 장치의 리소스의 일 예를 나타내는 도면이다.
- [0078] 도 5에서 IoT 플랫폼은 oneM2M 플랫폼이고, IoT 장치의 리소스는 flexContainer이다.
- [0079] 리소스는 프로토콜정보(nodnProtocol, nodnMappingInfo)를 자식 리소스 또는 속성으로서 갖는다.
- [0080] 예를 들어, IoT 장치의 프로토콜이 Modbus인 경우, IoT 장치의 리소스의 nodnProtocol에는 Modbus가 저장되고, nodnMappingInfo 속성에는 레지스터 파일 정보가 저장될 수 있다.
- [0082] 다음으로 도 6~도 9를 참조하여 커맨드 생성의 예시를 설명한다.
- [0083] 도 6은 Modbus 레지스터 파일을 나타내는 도면이다.
- [0084] 도 6을 참조하면, Modbus 레지스터 파일은 레지스터의 객체 타입(Coil, Discrete Inputs, Holding Registers, Input Register), 레지스터를 특정하기 위한 함수 코드(Function Code), 각 함수 코드에 해당하는 레지스터의 기능(Action)을 나타낸다.
- [0086] IoT 장치의 리소스를 생성하기 위해, IPE(Interworking Proxy application Entity)에 해당하는 리소스가 IoT 플랫폼(20)의 공통 서비스 엔터티에 생성된다. 그리고, IPE에 해당하는 리소스에 연결되어 관리되는 IoT 장치가 하부 리소스로 생성되고, IoT 장치의 센서가 IoT 장치에 해당하는 리소스의 하부 리소스로 생성된다. IoT 장치의 센서에 해당하는 리소스에 프로토콜 타입 및 맵핑 정보가 속성 값으로 저장될 수 있다.
- [0088] 도 7은 본 발명의 실시예에 따른 리소스를 나타내는 도면이다. 도 7의 (a)는 Modbus\_IPE 리소스, 도 7의 (b)는 deviceThermometer 리소스, 도 7의 (c)는 battery 리소스, 도 7의 (d)는 temperature 리소스를 나타낸다.
- [0089] Modbus\_IPE 리소스는 IPE에 해당하는 리소스이다. deviceThermometer 리소스는 IoT 장치에 해당하는 리소스이며, Modbus\_IPE 리소스의 자식 리소스이다. battery 리소스 및 temperature 리소스는 IoT 장치의 센서에 해당하는 리소스이며, deviceThermometer 리소스의 자식 리소스이다. battery 리소스 및 temperature 리소스는 프로토콜 타입(protocol) 및 맵핑 정보에 해당하는 레지스터 정보(nodnproperties)의 속성을 갖는다. deviceThermometer 리소스, battery 리소스 및 temperature 리소스는 flexContainer로 구현될 수 있다.
- [0091] 도 8은 본 발명의 실시예에 따른 리소스와 맵핑테이블의 대응을 설명하기 위한 도면이다.
- [0092] 도 8의 (a)는 battery 리소스의 레벨, 용량, 충전, 방전, 로우 배터리, 배터리문턱값, 전기에너지, 전압 및 재료의 속성을 갖는 경우 각 속성에 대한 정보(타입, Reable, Writable)를 나타내고, 도 7의 (b)는 맵핑테이블의 일부로서, 각 속성에 대한 Modbus 프로토콜 정보(Modbus 객체 타입, 필요한 레지스터의 개수, 주소)를 나타낸다.

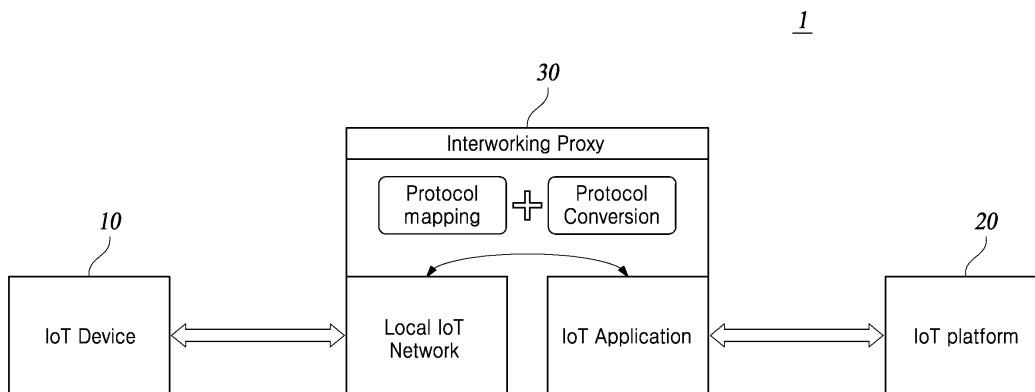
- [0093] 도 8의 (b)의 맵핑테이블은 도 6의 Modbus 레지스터 파일을 기초로 하여 생성된 것이다. 맵핑테이블에는 리소스 아이디가 저장되기 때문에, 리소스 생성시 리소스 아이디에 해당하는 맵핑테이블 항목의 프로토콜정보가 포함될 수 있다. 즉, 도 8의 (b)는 맵핑테이블의 일부를 도시한 것이지만, 동일한 내용이 리소스에 프로토콜정보로서 포함될 수 있다.
- [0095] 도 9는 도 7 및 도 8의 리소스를 이용하여 생성된, 배터리의 레벨을 읽어내기 위한 커맨드를 나타내는 도면이다.
- [0096] Slave ID는 배터리의 아이디를 나타낸다. 도 7에 도시되지는 않았지만 배터리의 아이디는 맵핑테이블 및 battery 리소스에 저장될 수 있다. 이에 따라, 리소스로부터 배터리의 아이디를 획득할 수 있다.
- [0097] 함수 코드는 도 6의 Modbus 레지스터 파일에서 도 8의 (b)의 Modbus 객체 타입인 Input Register에 대응하는 값이다. 맵핑테이블 또는 리소스에 저장된 객체 타입과, 레지스터 파일의 객체 타입에 해당하는 함수 코드를 참조함으로써 함수 코드를 생성되는 것으로 설명하였지만, 함수 코드는 맵핑테이블 또는 리소스에 직접 저장되고, 저장된 값을 직접 참조함으로써 생성될 수도 있다.
- [0098] 필요한 레지스터 개수 및 어드레스는 맵핑 테이블 및 리소스에 프로토콜 정보로서 저장된 값으로서 리소스로부터 얻을 수 있다.
- [0099] CRC(Cyclical Redundancy Check)는 커맨드의 오류를 체크하기 위해 추가되는 값이다.
- [0100] 이와 같이 생성된 커맨드를 배터리에 전송함으로써 프로키는 배터리로부터 배터리 레벨 값을 수신할 수 있다. 이후, 프로키는 수신된 값으로 배터리 레벨 리소스를 업데이트하게 된다.
- [0102] 도 10은 본 발명의 실시예에 따른 사물인터넷 네트워크(3)의 구성을 나타내는 도면이다.
- [0103] 도 10을 참조하면, 사물인터넷 네트워크(3)는 IoT 장치(10), IoT 플랫폼(20) 및 프로키(30)를 포함하며, 프로키(30)는, 미등록 장치(10)의 장치 아이디, 리소스 아이디 및 프로토콜 정보를 포함하는 맵핑테이블이 저장된 맵핑테이블 저장부(310); 상기 맵핑테이블에 기초하여, IoT 플랫폼에 상기 장치의 리소스를 생성하는 리소스 관리부(320); 및 상기 리소스에 기초하여, 상기 장치로부터 장치정보를 읽어내기 위한 커맨드를 생성하여 전송하고, 상기 커맨드에 대응하여 상기 장치로부터 장치정보를 수신하는 장치 통신부(330)를 포함하고, 상기 리소스 관리부(320)는, 상기 장치 통신부에서 수신된 장치정보를 이용하여 상기 리소스를 업데이트한다.
- [0105] 이상, 바람직한 실시예를 통하여 본 발명에 관하여 상세히 설명하였으나, 본 발명은 이에 한정되는 것은 아니며, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 다양하게 변경, 응용될 수 있음은 당해 기술분야의 통상의 기술자에게 자명하다. 따라서, 본 발명의 진정한 보호 범위는 다음의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술적 사상은 본 발명의 권리 범위에 포함되는 것으로 해석되어야 할 것이다.

도면

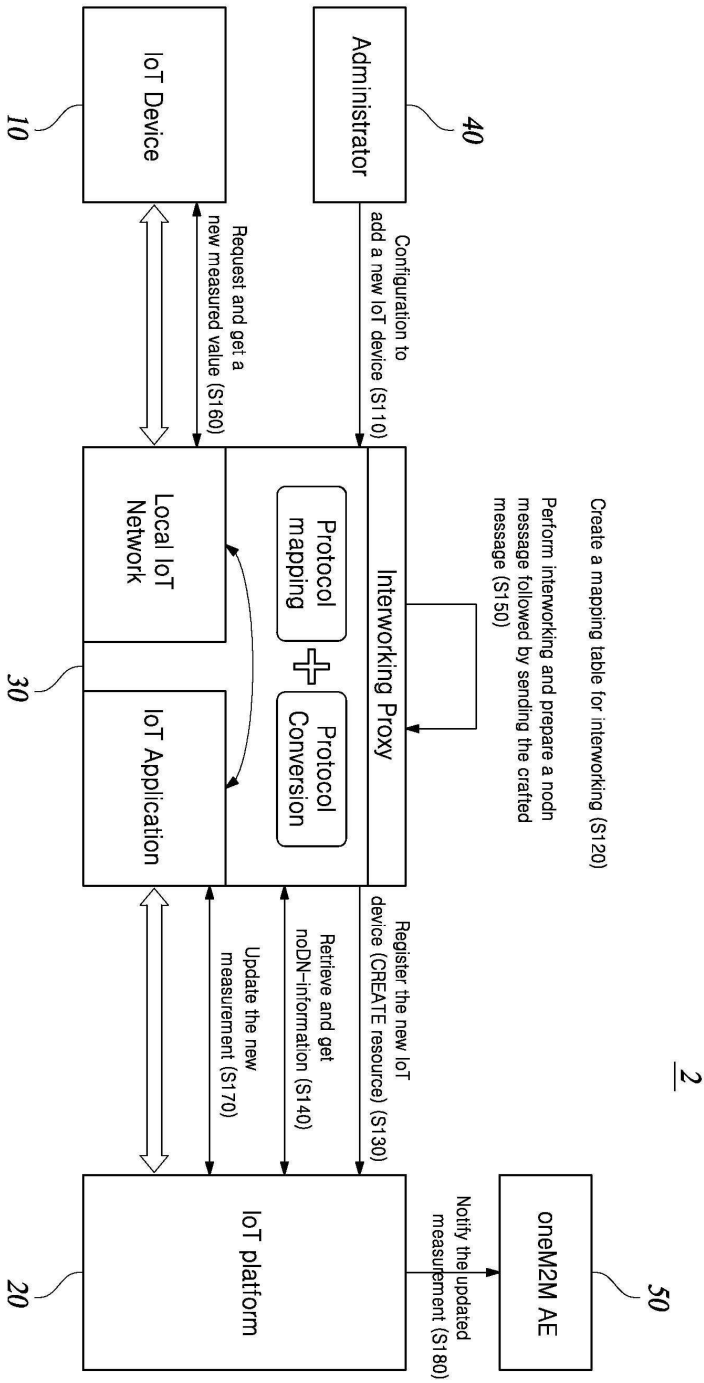
도면1



도면2



도면3

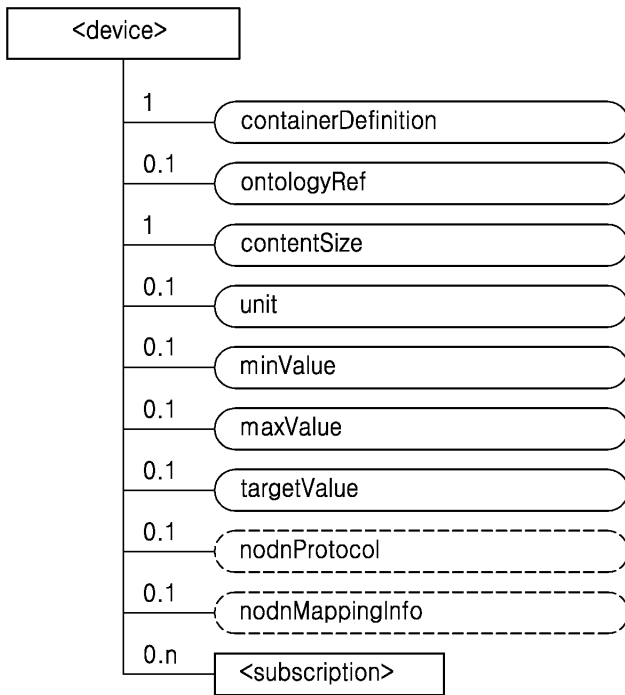


2

도면4

nodn#	nodn-ID	oneM2M-ID	nodnProtocol	createdTime	availability
1	Nodn-xxxxx1	<CSEBase>/flex1	Modbus	2019-10-03	O
2	Nodn-xxxxx2	<CSEBase>/flex2	Zigbee	2019-10-04	X

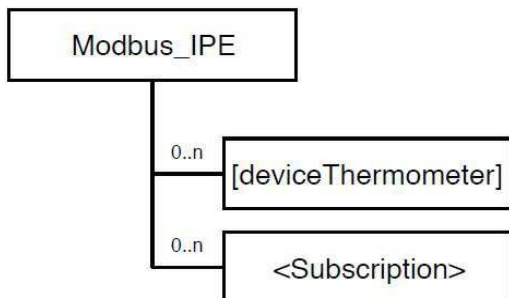
도면5



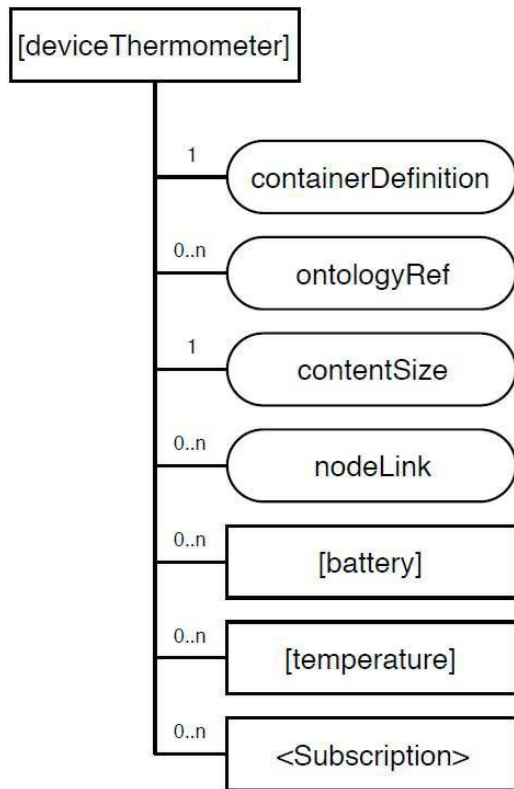
도면6

Register	Function Code	Action
Coil	01	Read
	05	Write(Single)
	0F	Write(Multiple)
Discrete Inputs	02	Read
Holding Registers	03	Read
	06	Write(Single)
	10	Write(Multiple)
Input Register	04	Read

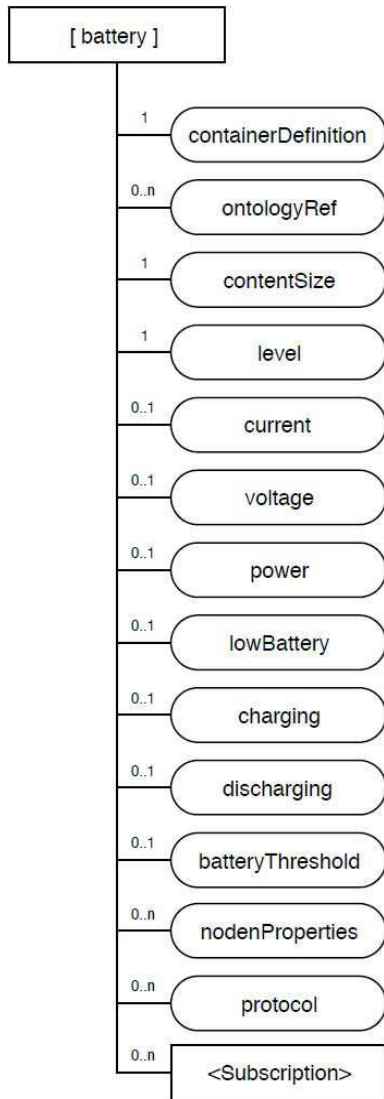
도면7a



도면7b

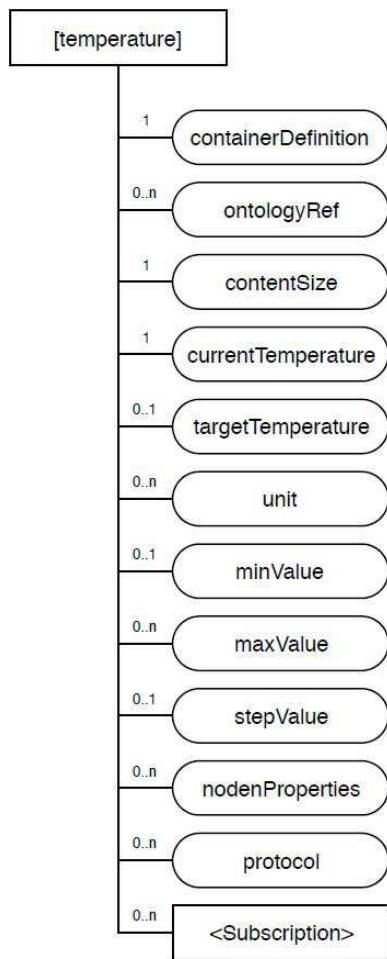


도면7c





도면7d



도면8

Battery Resource name	Type	Readable	Writable
level	xs:integer	true	false
capacity	xs:integer	true	false
charging	xs:boolean	true	false
discharging	xs:boolean	true	false
lowBattery	xs:boolean	true	false
batteryThreshold	xs:integer	true	true
electricEnergy	xs:integer	true	false
voltage	xs:integer	true	false
material	xs:string	true	false

(a)

Mapping

Modbus object type	Number of required registers	Address
Input Register(16-bit word), Read-Only	2 (32bit)	0x0001~0x0002
Input Register(16-bit word), Read-Only	2 (32bit)	0x0003~0x0004
Coil(Single bit), Read-Write	1 (8bit)	0x0001
Coil(Single bit), Read-Write	1 (8bit)	0x0002
Coil(Single bit), Read-Write	1 (8bit)	0x0003
Holding Register(16-bit word), Read-Write	2 (32bit)	0x0001~0x0002
Input Register(16-bit word), Read-Only	2 (32bit)	0x0005~0x0006
Input Register(16-bit word), Read-Only	2 (32bit)	0x0007~0x0008
Input Register(16-bit word), Read-Only	8 (128bit)	0x0009~0x0018

(b)

도면9

Slave ID	Function Code	Number of registers	Address	CRC (checksum)
01	04	0002	0001	200B

도면10

