KS X 3280

KSKSKSKS
KSKSKSK
KSKSKS
KSKSK
KSKS
KSK
KS

KS

에너지·전력분야 사물인터넷 인터페이스

KS X 3280:2021

방 송 통 신 표 준 심 의 회

2021년 11월 2일 제정

심 의 : 전파통신 기술심의회

	성 명	근 무 처	직 위
(회 장)	윤 영 중	연세대학교	교 수
(위 원)	김 기 형	아주대학교	교 수
	김 동 일	동의대학교	교 수
	김 창 주	한동대학교	교 수
	박 준 구	경북대학교	교 수
	송 평 중	한국전자통신연구원	전 문 위 원
	이 현 우	단국대학교	교 수
	최 상 호	전북테크노파크	센 터 장
	최 조 천	목포해양대학교	교 수
(간 사)	이 환 욱	과학기술정보통신부 국립전파연구원	과 장

원안작성협력 : 한국전력공사 전력연구원

	성 명	근 무 처	직 위
(연구책임자)	김 영 현	한국전력공사 전력연구원	책 임
(참여연구원)	명 노 길	한국전력공사 전력연구원	책 임
	강 동 훈	한국전력공사 전력연구원	선 임
	이 건 희	국가보안기술연구소	책 임
	김 용 운	한국전자통신연구원	실 장
	오 승 훈	한국전자통신연구원	책 임
	이 형 옥	한국전자통신연구원	선 임
	전 숙 현	한국정보통신기술협회	책 임
	서 창 수	㈜네오리플렉션	대 표
(간 사)	안 준 오	미래전파공학연구소	소 장

표준열람 : 국립전파연구원(http://www.rra.go.kr)

제 정 자 : 방송통신표준심의회 위원장　　담당부처 : 과학기술정보통신부 국립전파연구원
제　　정 : 2021년 11월 2일
심　　의 : 방송통신표준심의회 전파통신 기술심의회
원안작성협력 : 한국전력공사 전력연구원

이 표준에 대한 의견 또는 질문은 국립전파연구원 웹사이트를 이용하여 주십시오.

이 표준은 방송통신표준화지침 제18조의 규정에 따라 매 5년마다 방송통신표준심의회에서 심의되어 확인, 개정 또는 폐지됩니다.

목 차

머 리 말

이 표준은 방송통신발전기본법에 따라 방송통신표준심의회 심의를 거쳐 제정한 방송통신표준이다.

이 표준의 일부가 기술적 성질을 가진 특허권, 출원공개 이후의 특허출원, 실용신안권 또는 출원공개 후의 실용신안등록출원에 저촉될 가능성이 있다는 것에 주의를 환기한다. 관계 중앙행정기관의 장과 산업표준심의회는 이러한 기술적 성질을 가진 특허권, 출원공개 이후의 특허출원, 실용신안권 또는 출원공개 후의 실용신안등록출원에 관계되는 확인에 대하여 책임을 지지 않는다.

개 요

이 표준은 에너지 및 전력분야 사물인터넷 환경을 구성하기 위한 구성요소 간 상호운용성을 보장하기 위해 작성하였으며, 에너지·전력분야 사물인터넷(e-IoT)이 원활하게 운영되기 위해서는 플랫폼-게이트웨이, 게이트웨이-디바이스, 플랫폼-디바이스 간 상호운용성의 확보가 필요하다. 이러한 상호운용성 확보를 위하여 관련 용어 정의, 참조 모델, 서비스 절차, 정보 모델링, 보안 규격 등을 제시한다.

방송통신표준

KS X 3280:2021

에너지·전력분야 사물인터넷 인터페이스

Interface of internet of things in energy-electric power domain(e-IoT)

1 적용범위

이 표준은 에너지·전력분야의 현장에 구축될 사물인터넷 디바이스를 연결하여 디바이스에서 측정된 정보를 수집하고, 제어할 수 있으며 수집된 측정 데이터를 플랫폼에 전달할 수 있는 사물인터넷 정보모델링 및 연동규약을 정의하기 위한 것이다.

이 표준은 발전·송변전·배전 및 신재생에너지 분야 등의 에너지·전력 설비에 사용되는 다양한 기기들을 연결하고 정보를 수집·분석하는 서비스 제공을 위해서 플랫폼과 게이트웨이, 게이트웨이와 디바이스, 플랫폼과 디바이스 간 상호운용성을 확보할 수 있도록 단말등록 및 관리, 데이터 수집방법 등의 서비스 절차를 정의하고 상호운용성 확보를 위한 매개변수를 기술한다.

2 인용표준

다음의 인용표준은 전체 또는 부분적으로 이 표준의 적용을 위해 필수적이다. 발행연도가 표시된 인용표준은 인용된 판만을 적용한다. 발행연도가 표기되지 않은 인용표준은 최신판(모든 부록을 포함)을 적용한다.

IEC 62056, “Electricity metering data exchange — The DLMS/COSEM suite”
IEEE 2030.5, “Smart Energy Profile 2.0”
IETF RFC 4347, “Datagram Transport Layer Security”
IETF RFC 5280, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List(CRL) Profile”
IETF RFC 5288, “AES Galois Counter Mode (GCM) Cipher Suites for TLS”
IETF RFC 5289, “TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)”
IETF RFC 6209, “Addition of the ARIA(Academy, Research Institute, Agency) Cipher Suites to Transport Layer Security (TLS)”
IETF RFC 6347, “Datagram Transport Layer Security Version 1.2”
IETF RFC 6655, “AES-CCM Cipher Suites for Transport Layer Security”
IETF RFC 6690, “Constrained RESTful Environments(CoRE) Link Format”
IETF RFC 7251, “AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS”
IETF RFC 7252, “The Constrained Application Protocol(CoAP)”
IETF RFC 7641, “Observing Resources in CoAP”
IETF RFC 7959, “Block-wise-transfers in CoAP”
IETF RFC 8442, “ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites for TLS 1.2 and DTLS 1.2”
OMA_LwM2M_Core, “Lightweight Machine to Machine Technical Specification: Core Approved Version: 1.1.1”
OMA_LwM2M_Transport, “Lightweight Machine to Machine Technical Specification: Transport Bindings Approved Version:1.1.1”

KS X 1213 (2009), “128비트 블록암호 알고리즘 ARIA”
KS X 4105, “정보기술 — 개방형 시스템간 상호 접속 — 객체식별자(OID)의 구성과 등록 절차”

3 용어 정의 및 약어

이 표준의 목적을 위해 다음의 용어 정의를 적용한다.

3.1 용어 정의

3.1.1
사물인터넷(IoT, Internet of Things)
정보통신기술을 기반으로 실세계(physical world)와 가상 세계(virtual world)의 다양한 사물들을 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 시설을 말한다.

3.1.2
에너지·전력분야 사물인터넷(e-IoT,energy-IoT)
전력망의 곳곳에 설치된 다양한 장치들을 IoT기술로 연결하고 정보를 수집·분석하여 에너지·전력 서비스를 제공하는 시스템을 말한다.

3.1.3
e-IoT 디바이스(e-IoT Device)
e-IoT 시스템의 종단의 센서 또는 엑츄에이터 등과 같은 장치이다. 측정된 정보 데이터를 연결된 게이트웨이 또는 플랫폼에 표준 규격에 맞게 전달하는 역할을 수행하며, LwM2M 클라이언트 역할로 동작한다.

3.1.4
e-IoT 게이트웨이(e-IoT Gateway)
e-IoT 시스템의 디바이스로부터 정보를 수집하여 플랫폼으로 전달해 주는 장치이다. 여러 개의 디바이스와 통신하는 라우팅 기능을 지원할 수 있고 디바이스로부터 수집된 정보를 변환하여 전달하거나 우회하여 플랫폼에 전달하는 역할을 수행한다. 플랫폼과 연결 시에는 LwM2M 클라이언트 역할로 동작하며, 디바이스와 연결 시는 LwM2M 서버 역할로 동작한다.

3.1.5
e-IoT 플랫폼(e-IoT Platform)
e-IoT 시스템의 디바이스 및 게이트웨이로부터 전달받은 정보를 저장 및 가공할 수 있는 서버이다. 원격에서 디바이스와 게이트웨이를 제어할 수 있는 기능을 지원하며, LwM2M 서버 역할로 동작한다.

3.2 약어

AES	Advanced Encryption Standard
ARIA	Academy, Research Institute, Agency
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
IFgd	Interface between Gateway and Device
IFpd	Interface between Platform and Device

IFpg	Interface between Platform and Gateway
IPSO	Internet Protocol for Smart Objects
LwM2M	Lightweight Machine to Machine
OBJ-RES	Object Resource
OID	Object Identifier
OMA	Open Mobile Alliance
OMNA	OMA Naming Authority
OSI	Open Systems Interconnection
TLS	Transport Layer Security Protocol
URI	Uniform Resource Identifier

4 참조 모델

4.1 시스템 구성도

e-IoT 시스템은 종단의 에너지·전력 설비에 부착될 디바이스와 이들을 플랫폼에 연결해 주는 게이트웨이, 최종적으로 디바이스들의 모든 정보를 수집하고 디바이스들을 관리하여 다양한 e-IoT 서비스를 가능하게 해 주는 플랫폼으로 구성된다. 디바이스는 **그림 1**과 같이 게이트웨이를 통해서 플랫폼에 직접 연결될 수 있다.

e-IoT 시스템은 물리적 연결과 논리적 연결로 구분되며, 논리적 연결은 디바이스와 게이트웨이 간 인터페이스(IFgd), 게이트웨이와 플랫폼 간 인터페이스(IFpg), 디바이스와 플랫폼 간 인터페이스(IFpd)을 말하며 메시지 규격, 데이터 형식, 연동 절차, 오브젝트-리소스 정보모델, 보안 규격에 대해 정의한다.

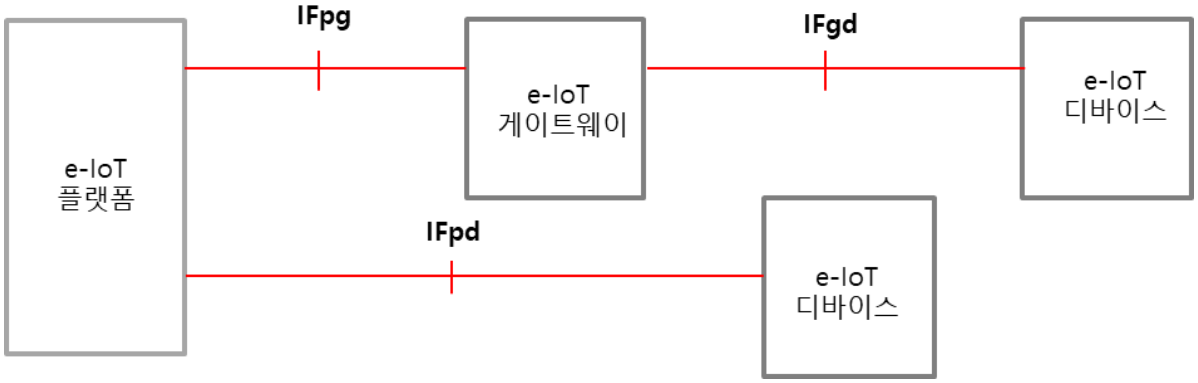

그림 1 — e-IoT 시스템 구성도

4.2 계층 모델

e-IoT 시스템 계층 모델은 OSI 계층 구조를 기본으로 한다.

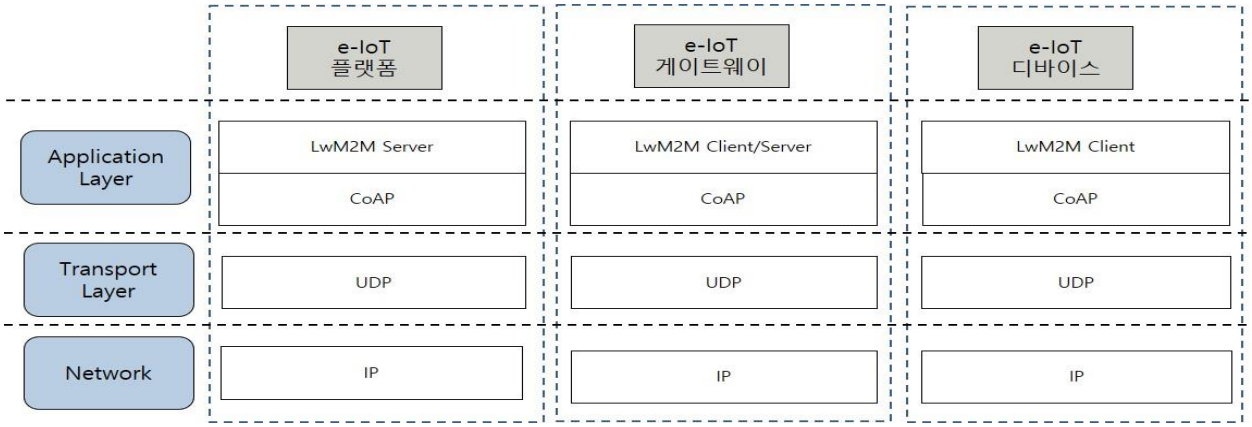

그림 2 — e-IoT 시스템 계층 모델

4.3 인터페이스 개념 정의

4.3.1 IFpg

e-IoT 시스템의 플랫폼과 게이트웨이간 메시지 교환을 위한 인터페이스이다.

4.3.2 IFgd

e-IoT 시스템의 게이트웨이와 디바이스간 메시지 교환을 위한 인터페이스이다.

4.3.3 IFpd

e-IoT 시스템의 플랫폼과 디바이스간 메시지 교환을 위한 인터페이스이다.

4.4 식별자

e-IoT 식별자는 디바이스와 게이트웨이를 식별하기 위해 사용되며, 유일성이 보장될 수 있도록 KS X 4105 표준 기반의 OID 체계를 따른다.

e-IoT 식별자는 **표 1**과 같이 구성되며 4번째 하위영역은 해당 기관에서 자체 규정에 따라 장치 식별자를 정의할 수 있다. 단 장치 식별자는 해당 기관 내에서 유일성이 보장되어야 한다.

표 1 — e-IoT 식별자 구성요소

<table>
<tr><th rowspan="2">영역</th><th colspan="4">식별자 구성 요소</th></tr>
<tr><th colspan="2">숫자형식</th><th colspan="2">이름형식</th></tr>
<tr><td>최상위 영역</td><td colspan="2">1</td><td colspan="2">iso</td></tr>
<tr><td>2번째 하위 영역</td><td colspan="2">2</td><td colspan="2">가맹기관(member-body)</td></tr>
<tr><td>3번째 하위 영역</td><td colspan="2">410</td><td colspan="2">한국(kor)</td></tr>
<tr><td rowspan="6">4번째 하위 영역</td><td colspan="2">0 ~ 9 999</td><td colspan="2">국내표준</td></tr>
<tr><td colspan="2">10 000 ~ 99 999</td><td colspan="2">장래의 확장을 위하여 확보</td></tr>
<tr><td rowspan="4">100 000 ~
16 099 999</td><td>100 000 ~ 100 999</td><td rowspan="4">기관</td><td>국가 기관</td></tr>
<tr><td>101 000 ~ 147 999</td><td>지방 정부 단체</td></tr>
<tr><td>148 000 ~ 199 999</td><td>예약 영역</td></tr>
<tr><td>200 000 ~ 16 099 999</td><td>상기 이외의 조직</td></tr>
</table>

5 e-IoT 인터페이스

5.1 플랫폼과 게이트웨이간 인터페이스(IFpg)

플랫폼과 게이트웨이 사이의 정보 교환을 위한 인터페이스로서, 게이트웨이 등록, 주기적 보고, 정보 조회, 제어, 펌웨어 업데이트 기능을 위해 사용된다. 특히, 등록 과정은 장치 기본등록과 단순등록 기능으로 나뉜다. 장치 기본등록 기능은 일반적인 등록 과정을 의미하며, 장치 단순등록 기능은 네트워크 대역폭이 작은 환경에서 단순화한 등록과정을 정의한다.

5.1.1 IFpg- 게이트웨이 기본 등록

본 절에서는 플랫폼이 게이트웨이와 디바이스의 오브젝트를 링크 포맷으로 등록하는 과정 및 각 오브젝트 별 리소스를 조회하는 절차에 대해서 정의한다.

그림 3과 같이 게이트웨이 기본 등록은 게이트웨이 오브젝트-리소스 등록과 디바이스 오브젝트-리소스 등록으로 각각 나누어서 수행된다. 게이트웨이는 디바이스를 대신하는 프록시 역할을 수행하며 게이트웨이와 디바이스의 등록하는 과정은 오브젝트를 등록하는 과정과 각 오브젝트별로 리소스를 조회하는 과정으로 구분된다. 리소스 조회 과정은 선택적으로 수행된다.

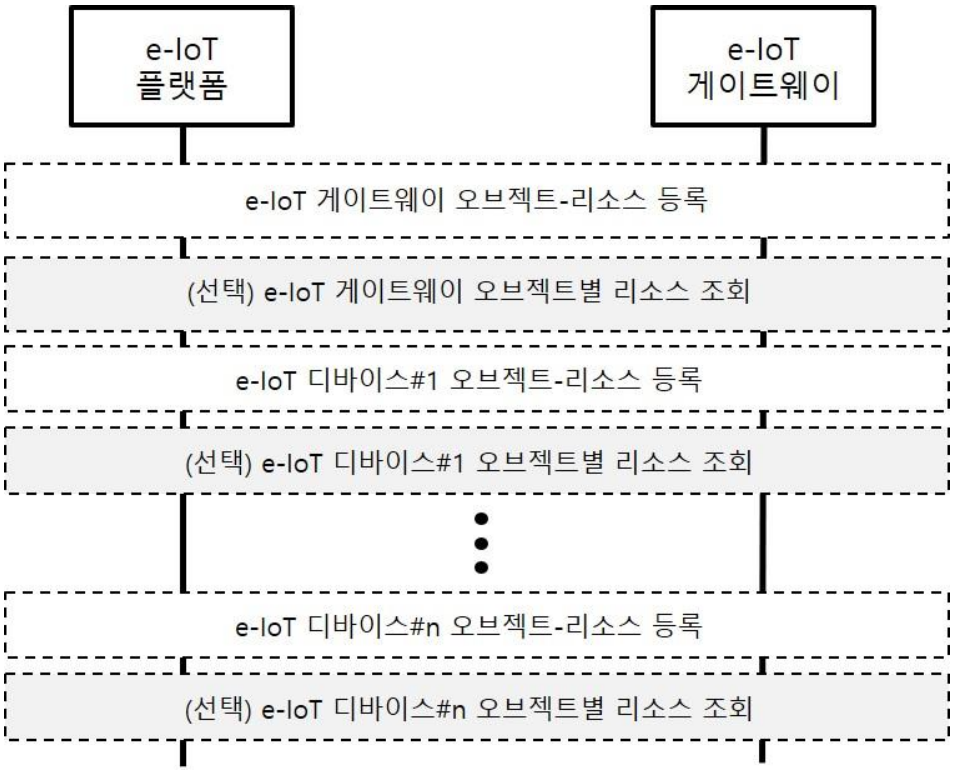

그림 3 — e-IoT 게이트웨이 기본 등록 절차

게이트웨이 기본 등록 과정을 수행한 결과 플랫폼에 등록된 게이트웨이의 데이터 모델의 형상은 **그림 4**와 같다. 게이트웨이 등록 시 전달되는 경로 정보와 디바이스 등록 시 전달되는 경로 정보가 각각 생성되며 그 하부에 오브젝트와 리소스 정보가 저장된다.

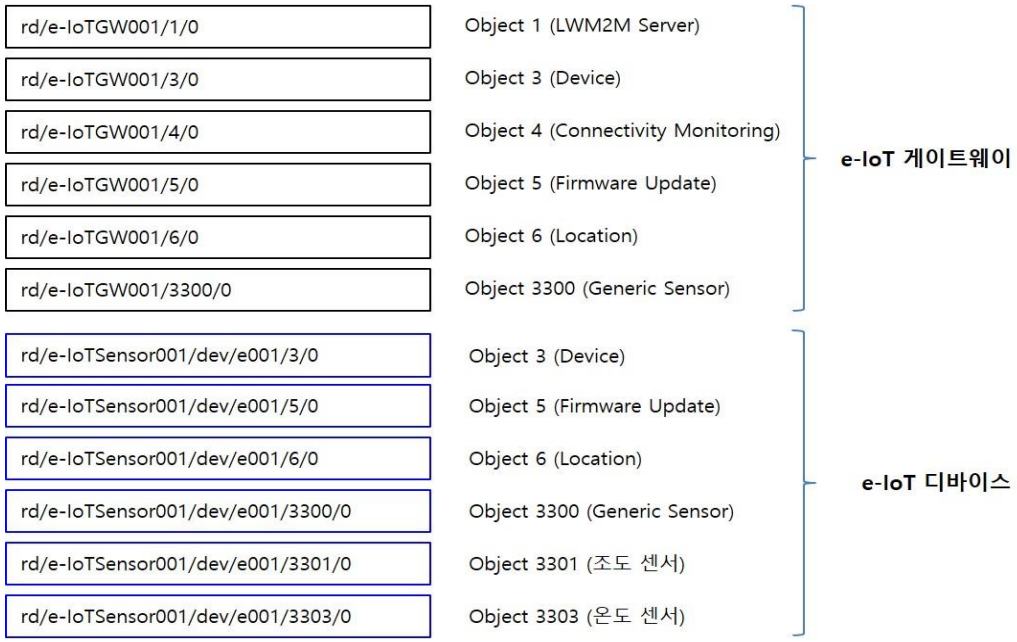

그림 4 — 플랫폼에 등록된 게이트웨이 데이터 모델

5.1.1.1 IFpg: 게이트웨이 기본 등록

게이트웨이 기본 등록 인터페이스는 **표 2**와 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 2 — 게이트웨이 기본 등록 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="14">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="10">URI</td><td rowspan="9">Template</td><td colspan="2" rowspan="2">—</td><td colspan="2">/{+rd}{?ep, lt, lwm2m, b, CK}</td></tr>
<tr><td>Uri-Path</td><td>rd</td></tr>
<tr><td colspan="2">Uri-Query</td><td>ep, lt, lwm2m, b, CK</td></tr>
<tr><td rowspan="6">변수</td><td>rd</td><td>필수</td><td colspan="2">리소스 디렉토리</td></tr>
<tr><td>ep</td><td>필수</td><td colspan="2">Endpoint Name(게이트웨이 OID)</td></tr>
<tr><td>lt</td><td>필수</td><td colspan="2">Lifetime</td></tr>
<tr><td>lwm2m</td><td>필수</td><td colspan="2">LwM2M 버전(1.0)</td></tr>
<tr><td>b</td><td>선택</td><td colspan="2">Binding Mode(U)</td></tr>
<tr><td>CK</td><td>선택</td><td colspan="2">시간 동기 요청</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 Location 헤더 포함</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td>2.01 Created</td><td rowspan="3">필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 Location 헤더 포함
— 요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td>4.00 Bad Request</td><td colspan="2" rowspan="2">등록 실패 메시지 수신한 경우, 게이트웨이는 60초 주기로 등록 요청 시도</td></tr>
<tr><td>4.03 Forbidden</td></tr>
</table>

5.1.1.2 IFpg: 디바이스 기본 등록

플랫폼에서 디바이스 기본 등록 기능은 게이트웨이가 대신 수행하며, 인터페이스는 **표 3**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 3 — 디바이스 기본 등록 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="12">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="9">URI</td><td colspan="2" rowspan="3">Template</td><td rowspan="3">—</td><td colspan="2">/{+rd}{?ep, lt, lwm2m, b, CK}</td></tr>
<tr><td>Uri-Path</td><td>rd</td></tr>
<tr><td>Uri-Query</td><td>ep, lt, lwm2m, b, CK</td></tr>
<tr><td rowspan="6">변수</td><td>rd</td><td>필수</td><td colspan="2">리소스 디렉토리</td></tr>
<tr><td>ep</td><td>필수</td><td colspan="2">Endpoint Name(디바이스 OID)</td></tr>
<tr><td>lt</td><td>필수</td><td colspan="2">Lifetime</td></tr>
<tr><td>lwm2m</td><td>필수</td><td colspan="2">LwM2M 버전(1.0)</td></tr>
<tr><td>b</td><td>선택</td><td colspan="2">Binding Mode(U)</td></tr>
<tr><td>CK</td><td>선택</td><td colspan="2">시간 동기 요청</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td colspan="2">디바이스의 정의된 오브젝트/오브젝트 인스턴스 포함, 디바이스의 Security 오브젝트는 포함하지 않음</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.01 Created</td><td rowspan="3">필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 Location 헤더 포함
— 요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">등록 실패 메시지 수신한 경우, 게이트웨이는 60초 주기로 등록 요청 시도</td></tr>
<tr><td colspan="2">4.03 Forbidden</td></tr>
</table>

5.1.1.3 IFpg: 게이트웨이 기본 등록 업데이트

게이트웨이의 기본 등록 업데이트 인터페이스는 **표 4**와 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 4 — 게이트웨이 기본 등록 업데이트 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="11">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="8">URI</td><td colspan="2" rowspan="3">Template</td><td rowspan="3">—</td><td colspan="2">/{+location}{?lt, b, CK}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>Uri-Query</td><td>lt, b, CK</td></tr>
<tr><td rowspan="4">변수</td><td>location</td><td>필수</td><td colspan="2">등록과정에서 게이트웨이가 수신한 경로정보</td></tr>
<tr><td>lt</td><td>선택</td><td colspan="2">Lifetime</td></tr>
<tr><td>b</td><td>선택</td><td colspan="2">Binding Mode(U)</td></tr>
<tr><td>CK</td><td>선택</td><td colspan="2">시간 동기 요청</td></tr>
<tr><td colspan="3">Content Format</td><td>선택</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="3">Payload</td><td>선택</td><td colspan="2">수정된 리소스 데이터가 있는 경우 포함</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="2">4.04 Not Found</td></tr>
</table>

5.1.1.4 IFpg: 디바이스 기본 등록 업데이트

플랫폼에서 디바이스 기본 등록 업데이트는 게이트웨이가 대신 수행하며, 인터페이스는 **표 5**와 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 5 — 디바이스 기본 등록 업데이트 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명	
요청 메시지	Transaction Type			필수	Confirmable	
	Method Type			필수	POST	
	URI	Template		—	/{+location}{?lt, b, CK}	
					Uri-Path	location
					Uri-Query	lt, b, CK
		변수	location	필수	등록과정에서 게이트웨이가 수신한 디바이스의 경로정보	
			lt	선택	Lifetime	
			b	선택	Binding Mode(U)	
			CK	선택	시간 동시 요청	
	Content Format			선택	application/link-format(40)	
	Payload			선택	수정된 리소스 데이터가 있는 경우 포함	
응답 메시지	응답 코드	2.04 Changed		필수	요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함	
		4.00 Bad Request			실패	
		4.04 Not Found				

5.1.1.5 IFpg: 게이트웨이 기본 등록 해제

게이트웨이 기본 등록 해제 인터페이스는 **표 6**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 6 — 게이트웨이 기본 등록 해제 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	DELETE
	URI	Template		—	/{location}
		변수	location	필수	등록과정에서 게이트웨이가 수신한 경로정보
응답 메시지	응답 코드	2.02 Deleted		필수	성공
		4.04 Not Found			실패

5.1.1.6 IFpg: 디바이스 기본 등록 해제

플랫폼에서 디바이스 기본 등록 해제는 게이트웨이가 대신 수행하며, 인터페이스는 **표 7**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 7 — 디바이스 기본 등록 해제 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	DELETE
	URI	Template		—	/{location}
		변수	location	필수	등록과정에서 게이트웨이가 수신한 디바이스의 경로정보
응답 메시지	응답 코드	2.02 Deleted		필수	성공
		4.04 Not Found			실패

5.1.2 IFpg- 게이트웨이 단순 등록

단순 등록은 장치 등록 과정을 단순화하여 사전에 오브젝트 리스트를 등록한 후 각 오브젝트 별로 정보를 조회하는 절차를 정의한다.

단순 등록은 반드시 사전에 기기 정보, 즉 오브젝트-리소스 프로파일 정보가 플랫폼에 저장되어 있어야 한다. 장치 단순 등록 과정 이후에는 별도의 오브젝트 정보 조회 과정을 생략할 수 있고 필요에 따라서 조회 과정을 수행할 수도 있다.

단순 등록 해제 절차는 기본 등록 해제 절차와 동일하다. 또한 단순 등록 이후에 기본 등록의 등록 업데이트 과정을 통해서 단순등록 정보를 유지한다. 실제 단순 등록 업데이트는 오브젝트-리소스 프로파일 수정을 통해서 이루어진다.

5.1.2.1 IFpg: 게이트웨이 단순 등록

플랫폼에서 게이트웨이 단순 등록을 위한 인터페이스는 **표 8**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 8 — 게이트웨이 단순 등록 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	POST
	URI	Template		—	/{+rd}{?lt, lep, GET, CK}
					Uri-Path: rd
					Uri-Query: lt, lep, GET, CK
		변수	rd	필수	리소스 디렉토리
			lt	필수	Lifetime
			lep	필수	Lightweight Endpoint Name(게이트웨이 OID)
			GET	선택	디바이스의 오브젝트-리소스 프로파일 정보 포함하여 응답 메시지 송부 요청
			CK	선택	시간 동기 요청, GET 옵션과 동시 사용 불가
	Content Format			선택	application/link-format(40)
	Payload			선택	대체 경로가 있는 경우 link-format 형식 포함
응답 메시지	응답 코드	2.01 Created		필수	— 등록된 오브젝트-리소스 정보에 해당하는 Location 헤더 포함 — 요청 메시지에 GET 포함 시, 페이로드에 해당 게이트웨이의 오브젝트-리소스 프로파일 포함 — 요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함
		4.00 Bad Request			실패
		4.03 Forbidden			
		4.04 Not Found			

5.1.2.2 IFpg: 게이트웨이 단순 등록 업데이트

게이트웨이 단순 등록 업데이트 절차는 다음과 같다.

- 플랫폼에서 수정된 게이트웨이의 오브젝트-리소스 프로파일이 존재할 경우 플랫폼이 게이트웨이에 재부팅 요청 메시지를 보낸다.
- 게이트웨이는 시스템 재부팅 후 **5.1.2.1** IFpg: 게이트웨이 단순 등록을 수행한다.

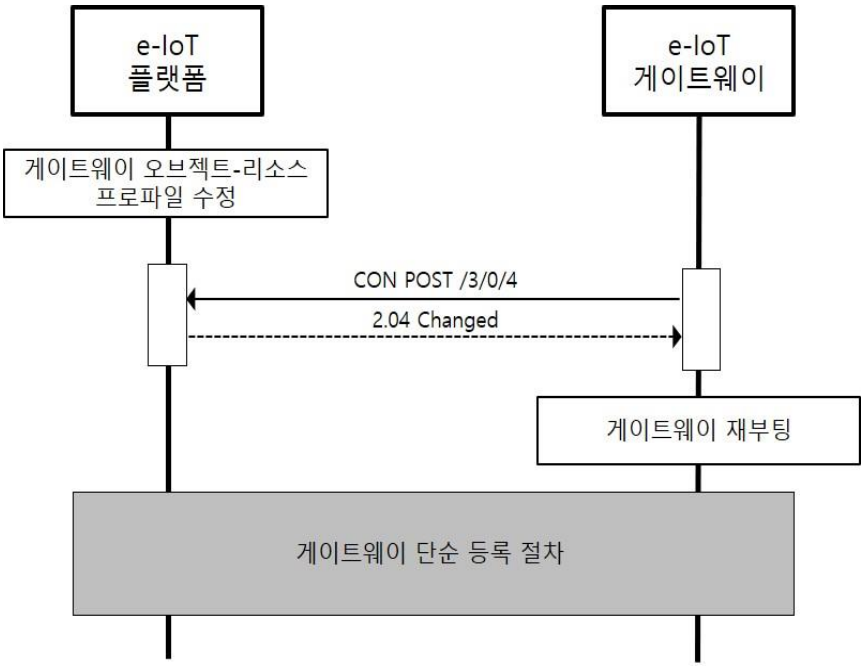

그림 5 — 게이트웨이 단순 등록 업데이트 절차

5.1.3 IFpg- 게이트웨이 주기적 보고

게이트웨이의 주기적 보고 기능은 보고 주기 속성값 설정, CoAP Observe 설정 및 주기적 보고, 주기 보고 해제 기능으로 구성된다.

5.1.3.1 IFpg: 보고 주기 속성값 설정

보고 주기 값에 해당하는 대상 리소스의 pmax 속성을 설정하는 절차로서 게이트웨이에 연결된 모든 디바이스의 측정값이 플랫폼에 보고되어야 하므로 보고 주기값 설정이 필요한 모든 리소스에 본 과정이 적용된다.

게이트웨이 보고 주기 및 관련 속성값 설정을 위한 인터페이스는 **표 9**와 같이 요청 및 응답 메시지로 정의되며 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 9 — 주기적 보고 속성값 설정 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="6">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">PUT</td></tr>
<tr><td rowspan="4">URI</td><td colspan="2" rowspan="3">Template</td><td rowspan="3">—</td><td colspan="2">/{location}{?pmax}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>Uri-Query</td><td>pmax</td></tr>
<tr><td rowspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td>pmax</td><td>필수</td><td colspan="2">보고 주기값</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="2">4.04 Not Found</td></tr>
</table>

5.1.3.2 IFpg: CoAP Observe 설정 및 주기적 보고

대상 오브젝트와 리소스에 Observe 속성 설정을 통한 주기적 보고 절차를 정의한다. Observe 설정 대상이 리소스인 경우 주기적 보고가 필요한 모든 리소스에 이 과정을 적용된다.

게이트웨이 주기적 보고를 위한 Observe 속성 설정 인터페이스는 **표 10**과 같이 요청 및 응답 메세지로 정의되며 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 10 — CoAP Observe 설정 및 주기적 보고 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="7">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">GET</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td colspan="2">location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td colspan="4">Observe</td><td>필수</td><td colspan="2">0</td></tr>
<tr><td colspan="4">Accept</td><td>필수</td><td colspan="2">— Observe 대상이 리소스인 경우: Plain-text 또는 JSON
— Observe 대상이 오브젝트 인스턴스인 경우: JSON</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td rowspan="2">2.05 Content</td><td>Observe</td><td rowspan="3">필수</td><td rowspan="3">성공</td><td>0보다 큰 값</td></tr>
<tr><td>Content Format</td><td>요청 메시지의 Accept와 동일</td></tr>
<tr><td>Payload</td><td>해당 리소스의 값</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td colspan="2">실패</td></tr>
</table>

5.1.3.3 IFpg: 주기 보고 해제

게이트웨이의 주기 보고 해제를 위한 Observe 속성 설정 인터페이스는 **표 11**과 같이 요청 및 응답 메시지로 정의되며 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 11 — 주기적 보고 해제 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="6">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">GET</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td colspan="4">Observe</td><td>필수</td><td colspan="2">1</td></tr>
<tr><td colspan="4">Accept</td><td>필수</td><td colspan="2">— Observe 대상이 리소스인 경우: Plain-text 또는 JSON
— Observe 대상이 오브젝트 인스턴스인 경우: JSON</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td>2.05 Content</td><td colspan="2">Content Format
Payload</td><td rowspan="3">필수</td><td rowspan="2">성공</td><td>요청 메시지의 Accept와 동일
해당 리소스의 값</td></tr>
<tr><td colspan="3">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="3">4.04 Not Found</td></tr>
</table>

5.1.4 IFpg- 게이트웨이 단순 주기적 보고

게이트웨이가 플랫폼에 주기적 보고 시 대상 오브젝트-리소스의 snot(Simple Notification) 속성이 True인 경우 단순 주기적 보고를 수행한다. 이 경우 미리 설정된 pmax 주기로 해당 리소스 값을 주기적으로 전송한다. 주기적 보고 대상이 되는 리소스의 최종 경로는 게이트웨이 등록 시 획득한 경로(Location-Path) 정보와 사전에 지정한 리소스 경로의 조합으로 결정된다.

게이트웨이의 단순 주기적 보고 인터페이스는 **표 12**와 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자 역할을 한다.

표 12 — 게이트웨이 단순 주기적 보고 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="7">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable 또는
Non-Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">PUT</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td colspan="4">Content Format</td><td>필수</td><td colspan="2">Plain-text(0)</td></tr>
<tr><td colspan="4">Payload</td><td>필수</td><td colspan="2">대상 리소스의 값</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="3">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="3">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="3">4.04 Not Found</td></tr>
</table>

5.1.5 IFpg- 게이트웨이 정보 조회

플랫폼이 게이트웨이의 정보를 조회하는 기능은 해당 오브젝트와 리소스의 속성값을 읽어오는 Read 동작과 해당 오브젝트에 구현된 리소스를 조회하는 Discovery 동작으로 구분된다.

5.1.5.1 IFpg: 정보 조회 - Read

게이트웨이의 Read 동작을 통한 정보 조회 인터페이스는 **표 13**과 같이 요청 및 응답 메세지로 정의되며 플랫폼은 요청자, 플랫폼은 응답자 역할을 한다.

표 13 — 정보 조회(Read) 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th>정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td>Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td>GET</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td>/{location}</td></tr>
<tr><td>Uri-Path | location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td>대상 리소스의 URI</td></tr>
<tr><td colspan="4">Accept</td><td>필수</td><td>Plain-text(0) 또는 JSON(11543)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td>2.05 Content</td><td colspan="2">Content Format
Payload</td><td rowspan="3">필수</td><td>요청 메시지의 Accept와 동일
해당 리소스의 값</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td>실패</td></tr>
</table>

5.1.5.2 IFpg: 정보 조회 - Discovery

게이트웨이의 Discovery 동작을 통한 정보 조회 인터페이스는 **표 14**와 같이 요청 및 응답 메세지로 정의되며 이 경우 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 14 — 정보 조회(Discovery) 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th>정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td>Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td>GET</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td>/{location}</td></tr>
<tr><td>Uri-Path | location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td>대상 리소스의 URI</td></tr>
<tr><td colspan="4">Accept</td><td>필수</td><td>application/link-format(40)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td>2.05 Content</td><td colspan="2">Content Format
Payload</td><td rowspan="3">필수</td><td>요청 메시지의 Accept와 동일
오브젝트/오브젝트 리스트/리소스 정보</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td>실패</td></tr>
</table>

5.1.6 IFpg- 게이트웨이 제어

플랫폼이 게이트웨이를 제어하는 기능은 대상 리소스에 값을 설정하여 쓰기(Write) 또는 실행(Execute) 동작을 목적으로 한다.

게이트웨이의 제어 기능 인터페이스는 **표 15**와 같이 요청 및 응답 메시지로 정의되며 이 경우 플랫폼은

요청자, 게이트웨이는 응답자 역할을 한다.

표 15 — 제어(쓰기 및 실행) 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th>정의 및 설명</th></tr>
<tr><td rowspan="8">요청 메시지</td><td colspan="3">Transaction Type</td><td></td><td>필수</td><td>Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td></td><td>필수</td><td>PUT(쓰기) 또는 POST(실행)</td></tr>
<tr><td rowspan="3">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2">—</td><td>/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td>location</td><td>필수</td><td>대상 리소스의 URI</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td>Plain-text(0) 또는 JSON(11543)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td>쓰기 또는 실행하고자 하는 값</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.04 Changed</td><td rowspan="3">필수</td><td>성공</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td>실패</td></tr>
<tr><td colspan="2">4.04 Not Found</td></tr>
</table>

5.1.7 IFpg- 게이트웨이 펌웨어 업데이트

플랫폼과 게이트웨이 간 펌웨어 업데이트 절차는 펌웨어 파일 전달, 다운로드 상태 확인, 업데이트 수행, 재등록 및 업데이트 결과 확인 과정으로 진행된다.

5.1.7.1 IFpg: 펌웨어 파일 전달

게이트웨이의 펌웨어 파일 전달을 위한 인터페이스는 **표 16**과 같이 요청 및 응답 메시지로 정의되며 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 16 — 펌웨어 파일 전달 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th>정의 및 설명</th></tr>
<tr><td rowspan="8">요청 메시지</td><td colspan="3">Transaction Type</td><td></td><td>필수</td><td>Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td></td><td>필수</td><td>PUT</td></tr>
<tr><td rowspan="3">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2">—</td><td>/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td>location</td><td>필수</td><td>펌웨어 업데이트 오브젝트 - 패키지 리소스의 URI (5/0/0)</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td>octet-stream(42)</td></tr>
<tr><td colspan="3">Block1</td><td>필수</td><td>펌웨어파일이 최대전송단위(MTU)보다 큰 경우
— NUM: 0 보다 큰 정수(매전송마다 증가)
— M: 1(파일분할 계속), 0(파일분할 마지막)
— SZX: Payload 크기 = 2(SZX+4)로 계산됨</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td>펌웨어 파일 블록</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="2">응답 코드</td><td>2.31 Continue</td><td>Block1</td><td rowspan="2">필수</td><td>파일 분할 계속</td></tr>
<tr><td>2.04 Changed</td><td>Block1</td><td>파일 분할 마지막</td></tr>
<tr><td colspan="5">— Message ID는 요청 메시지의 것과 동일하게 설정
— Token는 Block Transfer과정이 끝날 때까지 첫 요청 메시지와 동일한 값으로 설정
— Block1 옵션을 요청 메시지의 것과 동일한 것으로 설정</td></tr>
</table>

5.1.7.2 IFpg: 펌웨어 파일 다운로드 상태 확인

게이트웨이의 펌웨어 파일 다운로드 상태확인을 위한 인터페이스는 **표 17**과 같이 요청 및 응답 메시지로 정의되며 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 17 — 펌웨어 파일 다운로드 상태 확인 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">GET</td></tr>
<tr><td rowspan="2">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">펌웨어 업데이트 오브젝트 - 상태 리소스의 URI (5/0/3)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td rowspan="2">2.05 Content</td><td colspan="2">Content Format</td><td rowspan="3">필수</td><td colspan="2">Plain-text(0)</td></tr>
<tr><td colspan="2">Payload</td><td colspan="2">다운로드 상태값
(0: Idle, 1: Downloading, 2: Downloaded, 3: updating)</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td colspan="2">실패</td></tr>
</table>

5.1.7.3 IFpg: 펌웨어 업데이트 수행

플랫폼이 게이트웨이의 펌웨어 파일 다운로드가 성공적으로 완료되었는지 확인 후 펌웨어 업데이트 수행을 위한 인터페이스는 **표 18**과 같이 요청 및 응답 메시지로 정의되며 이 경우 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 18 — 펌웨어 업데이트 수행 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="2">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">펌웨어 업데이트 오브젝트 - 업데이트 리소스의 URI (5/0/2)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="3">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="3">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="3">4.04 Not Found</td></tr>
</table>

5.1.7.4 IFpg: 재등록 및 업데이트 결과 확인

게이트웨이는 성공적으로 펌웨어 업데이트가 완료되면 게이트웨이 등록과정을 수행한다. 관련 절차는 **5.1.1.1** IFpg: 게이트웨이 기본 등록 과정을 참고한다.

게이트웨이가 등록 과정을 완료 후 플랫폼이 펌웨어 업데이트 결과 확인을 위한 인터페이스는 **표 19**와 같이 요청 및 응답 메시지로 정의되며 이 경우 플랫폼은 요청자, 게이트웨이는 응답자 역할을 한다.

표 19 — 펌웨어 업데이트 결과 확인 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="3">Transaction Type</td><td></td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td></td><td>필수</td><td colspan="2">GET</td></tr>
<tr><td rowspan="3">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2"></td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td colspan="2">변수</td><td>location</td><td>필수</td><td colspan="2">펌웨어 업데이트 오브젝트 – 업데이트 결과 리소스의 URI (5/0/5)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td rowspan="2">2.05 Content</td><td colspan="2">Content Format</td><td rowspan="3">필수</td><td colspan="2">Plain-text(0)</td></tr>
<tr><td colspan="2">Payload</td><td colspan="2">펌웨어 업데이트 결과값
(0: 초기값, 1: 성공, 2: 새 펌웨어 패키지 저장 공간 부족, 3: 다운로드 중 저장공간 부족, 4: 다운로드 중 네트워크 연결실패, 5: 새 펌웨어 패키지 무결성 검증 실패, 6: 미지원 펌웨어 패키지 타입, 7: 비유효한 URI, 8: 펌웨어 업데이트 실패)</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td colspan="2">실패</td></tr>
</table>

5.2 게이트웨이와 디바이스간 인터페이스(IFgd)

게이트웨이와 디바이스 사이의 정보 교환을 위한 인터페이스로서, 디바이스 등록, 주기적 보고, 단순 주기적 보고, 정보 조회 및 제어 기능을 위해 사용된다. 특히 등록 과정은 디바이스 기본 등록과 단순 등록 기능으로 나뉜다.

5.2.1 IFgd- 디바이스 기본 등록

디바이스 기본 등록은 게이트웨이의 리소스 디렉토리 검색, 디바이스 등록, 등록 업데이트 및 등록 해제 기능으로 구분된다. 기본 등록 관련 기능은 디바이스의 물리적인 제약사항으로 인해 **표 20**과 같이 선택과 필수 기능으로 정의한다.

표 20 — 디바이스 등록 관련 기능

	등록 관련 기능	필수/선택
1	리소스 디렉토리 검색	선택
2	디바이스 기본 등록	필수
3	디바이스 기본 등록 업데이트	필수
4	디바이스 기본 등록 해제	선택

5.2.1.1 IFgd: 리소스 디렉토리 검색

디바이스가 게이트웨이의 리소스 디렉토리를 검색하기 위해서는 게이트웨이의 IP 주소, 포트 번호, 리소스 디렉토리 경로 정보를 알아야 한다. 만약 디바이스가 리소스 디렉토리 검색 기능을 지원하지 않는 경우 관련 정보를 미리 설정해야 한다.

디바이스의 게이트웨이 리소스 디렉토리 검색 인터페이스는 **표 21**과 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 21 — 게이트웨이 리소스 디렉토리 검색 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	GET
	URI	Template		—	/.well-known/core{?rt}
		변수	rt	선택	리소스 타입(“core.rd”, “core.rd=lookup”, “core.rd-group”, “core.rd*” 중 하나)
	Accept			필수	application/link-format(40)
응답 메시지	응답 코드	2.05 Content	Content Format Payload	필수	요청 메시지의 Accept와 동일
					리소스 디렉토리 정보
		4.00 Bad Request			실패
		4.04 Not Found			

5.2.1.2 IFgd: 디바이스 기본 등록

디바이스의 기본 등록 인터페이스는 **표 22**와 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 22 — 디바이스 기본 등록 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명	
요청 메시지	Transaction Type			필수	Confirmable	
	Method Type			필수	POST	
	URI	Template		—	/{+rd}{?ep, lt, CK}	
					Uri-Path	rd
					Uri-Query	ep, lt, CK
		변수	rd	필수	리소스 디렉토리	
			ep	필수	Endpoint Name(디바이스 OID)	
			lt	필수	Lifetime	
			CK	선택	시간 동기 요청	
	Content Format			필수	application/link-format(40)	
	Payload			필수	디바이스에 정의된 오브젝트와 오브젝트 인스턴스 정보	
응답 메시지	응답 코드	2.01 Created		필수	— 등록된 오브젝트-리소스 정보에 해당하는 Location-Path 헤더 포함 — 요청 메시지에 CK 포함 시 페이로드에 게이트웨이의 시스템 시간 포함	
		4.00 Bad Request			실패	
		5.03 Service Unavailable				

그림 6은 디바이스가 게이트웨이에 기본 등록 절차 완료 후 플랫폼에 기본 등록되는 전체 과정이다. 게이트웨이의 플랫폼 기본등록 과정은 본 문서의 **5.1.1.2** IFpg: 디바이스 기본 등록을 참고한다.

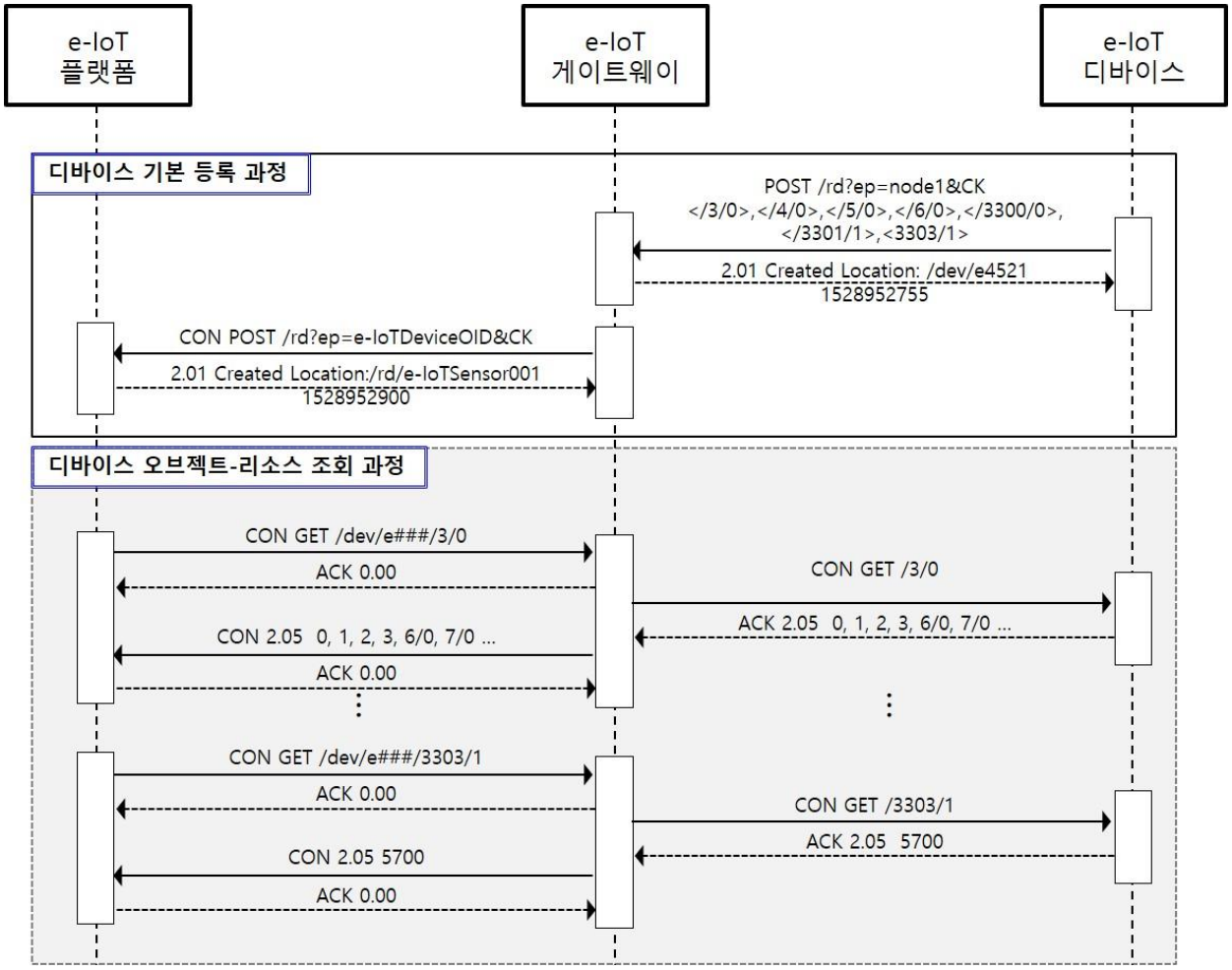

그림 6 — 디바이스 기본등록 전체 절차

5.2.1.3 IFgd: 디바이스 기본 등록 업데이트

디바이스의 기본 등록 업데이트를 위한 인터페이스는 **표 23**과 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 23 — 디바이스 기본 등록 업데이트 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명	
요청 메시지	Transaction Type			필수	Confirmable	
	Method Type			필수	POST	
	URI	Template		—	/{+location}{?lt, CK}	
					Uri-Path	location
					Uri-Query	lt, CK
		변수	location	필수	등록과정에서 디바이스가 수신한 경로정보	
			lt	선택	Lifetime	
			CK	선택	시간 동기 요청	
	Content Format			선택	application/link-format(40)	
	Payload			선택	수정된 리소스 데이터가 있는 경우 포함	

메시지 종류	속성		필수 여부	정의 및 설명
응답 메시지	응답 코드	2.04 Changed	필수	성공, 요청 메시지에 CK 포함 시, 페이로드에 게이트웨이의 시스템 시간 포함
		4.00 Bad Request		실패
		4.04 Not Found		
		5.03 Service Unavailable		

그림 7은 디바이스가 게이트웨이에 기본 등록 업데이트 후 플랫폼에 업데이트 되는 전체 과정이다. 단, IFgd 인터페이스의 디바이스 기본 등록 업데이트와 IFpg 인터페이스의 디바이스 기본 등록 업데이트는 독립적으로 수행한다. 게이트웨이와 플랫폼 간 디바이스 등록 업데이트 과정은 본 문서의 **5.1.1.4** IFpg: 디바이스 기본 등록 업데이트를 참고한다.

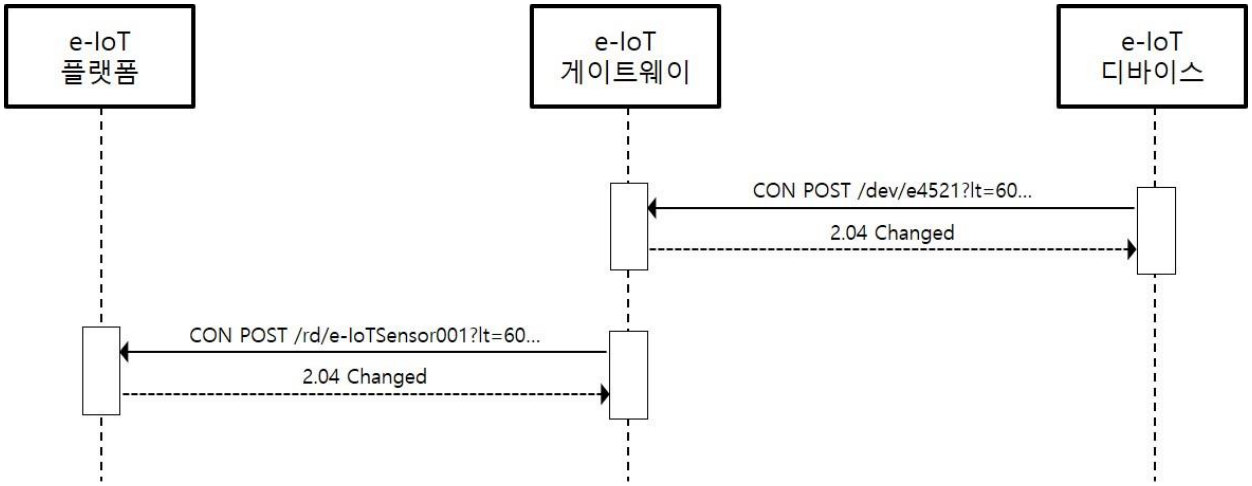

그림 7 — 디바이스 등록 업데이트 전체 절차

5.2.1.4 IFgd: 디바이스 기본 등록 해제

디바이스의 기본 등록 해제 인터페이스는 **표 24**와 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 24 — 디바이스 기본 등록 해제 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명	
요청 메시지	Transaction Type			필수	Confirmable	
	Method Type			필수	DELETE	
	URI	Template		—	/{location}	
					Uri-Path	location
		변수	location	필수	등록과정에서 디바이스가 수신한 경로정보	
응답 메시지	응답 코드	2.02 Deleted		필수	성공	
		4.04 Not Found			실패	

그림 8은 디바이스가 게이트웨이에 기본 등록 해제 완료 후 플랫폼에 기본 등록 해제되는 전체 과정이다.

게이트웨이와 플랫폼 간 디바이스 등록 해제 과정은 본 문서의 **5.1.1.6** IFpg: 디바이스 등록 해제를 참고한다.

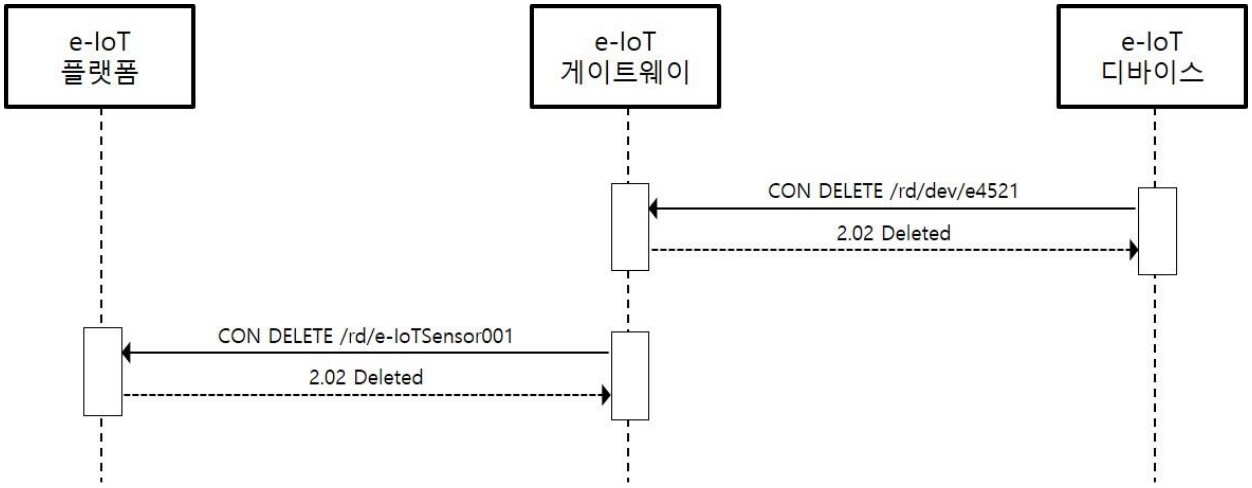

그림 8 — 디바이스 등록 해제 전체 절차

5.2.2 IFgd- 디바이스 단순 등록

디바이스 단순 등록은 반드시 사전에 디바이스의 오브젝트-리소스 프로파일 정보가 플랫폼에 저장되어 있어야 하며, 등록 요청 메시지에 lep(Lightweight Endpoint Name) 옵션을 포함해야 한다.

게이트웨이는 디바이스의 오브젝트-리소스 프로파일 정보가 필요할 경우, 플랫폼에 GET 옵션을 포함한 단순 등록 과정을 수행하여 관련 정보를 획득할 수 있다.

5.2.2.1 IFgd: 디바이스 단순 등록

디바이스 단순 등록을 위한 인터페이스는 **표 25**과 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 25 — 디바이스 단순 등록 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th>정의 및 설명</th></tr>
<tr><td rowspan="6">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td>Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td>POST</td></tr>
<tr><td rowspan="4">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2">—</td><td>/{+rd}{?lep}</td></tr>
<tr><td>Uri-Path | rd</td></tr>
<tr><td rowspan="2">변수</td><td>rd</td><td>필수</td><td>리소스 디렉토리</td></tr>
<tr><td>lep</td><td>필수</td><td>Lightweight Endpoint Name(디바이스 OID)</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.01 Created</td><td rowspan="3">필수</td><td>등록된 오브젝트-리소스 정보에 해당하는 경로 정보(Location-Path) 헤더 포함</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td rowspan="2">실패</td></tr>
<tr><td colspan="2">5.03 Service Unavailable</td></tr>
</table>

게이트웨이가 디바이스의 오브젝트-리소스 정보가 없는 경우 플랫폼에 디바이스 단순 등록을 요청하는 인터페이스는 **표 26**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 플랫폼은 응답자

역할을 한다.

표 26 — 게이트웨이의 디바이스 단순 등록 요청 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="8">요청 메시지</td><td colspan="2">Transaction Type</td><td></td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="2">Method Type</td><td></td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="4">URI</td><td rowspan="4">Template</td><td rowspan="4">—</td><td colspan="2">/{+rd}{?lep}</td></tr>
<tr><td>Uri-Path</td><td>rd</td></tr>
<tr><td>Uri-Query</td><td>lep, GET</td></tr>
<tr><td rowspan="3">변수</td><td>rd</td><td>필수</td><td colspan="2">리소스 디렉토리</td></tr>
<tr><td>lep</td><td>필수</td><td colspan="2">Lightweight Endpoint Name(디바이스 OID)</td></tr>
<tr><td>GET</td><td>선택</td><td colspan="2">오브젝트-리소스 프로파일 요청시</td></tr>
<tr><td colspan="2">Content Format</td><td></td><td>필수</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="2">Payload</td><td></td><td>선택</td><td colspan="2">게이트웨이가 생성한 디바이스의 경로정보</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td>2.01 Created</td><td rowspan="3">필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 경로 정보(Location-Path) 헤더 포함
— 페이로드에 디바이스 오브젝트-리소스 프로파일 포함(JSON(11543))</td></tr>
<tr><td>4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td>5.03 Service Unavailable</td></tr>
</table>

그림 9는 디바이스가 게이트웨이에 단순 등록 절차 완료 후 플랫폼에 단순 등록되는 전체 과정이다.

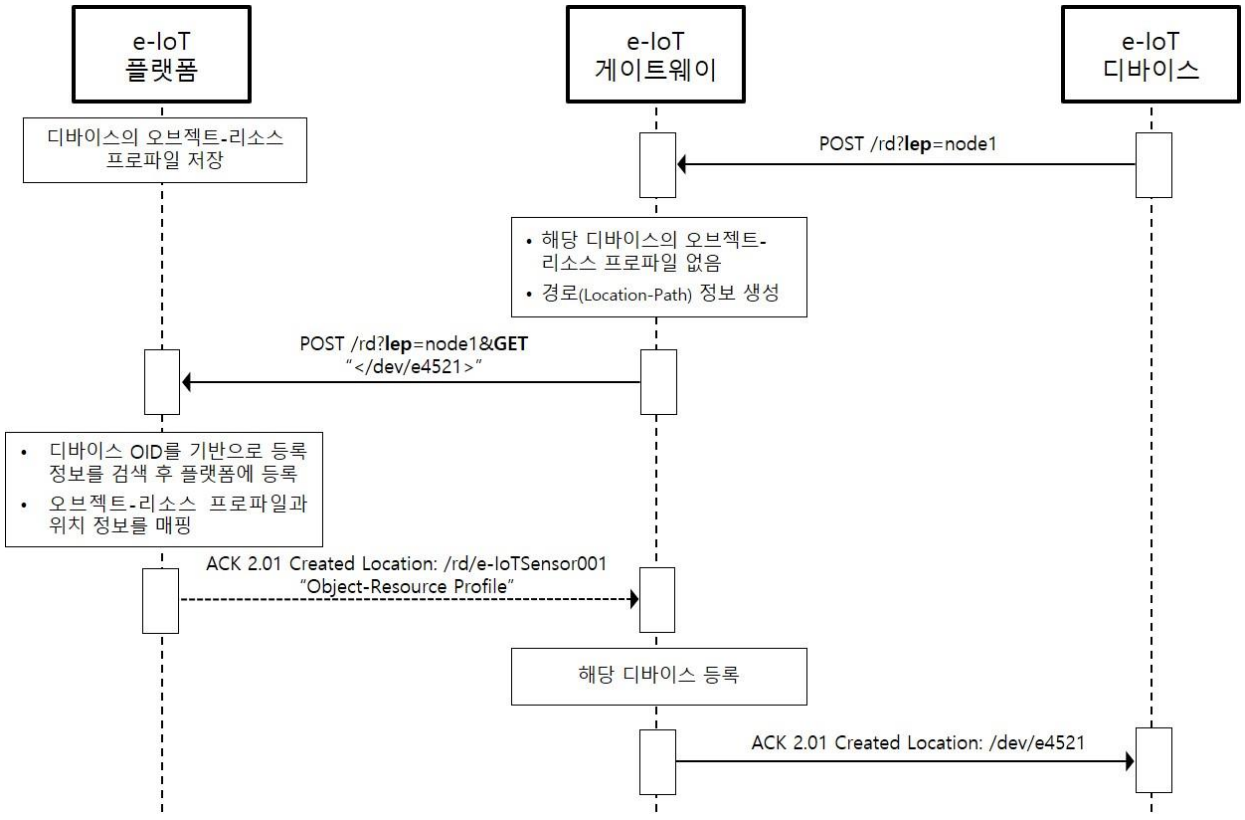

그림 9 — 디바이스 단순 등록 전체 절차

5.2.2.2 IFgd: 디바이스 단순 등록 업데이트

디바이스 단순 등록 업데이트 절차는 다음과 같다.

- 플랫폼에서 수정된 디바이스 오브젝트-리소스 프로파일이 존재할 경우 플랫폼이 게이트웨이에 해당 디바이스의 오브젝트-리소스 프로파일을 삭제 요청 및 디바이스 재부팅 요청 메시지를 보낸다.
- 디바이스는 시스템 재부팅 후 **5.2.2.1** IFgd: 디바이스 단순 등록을 수행한다.

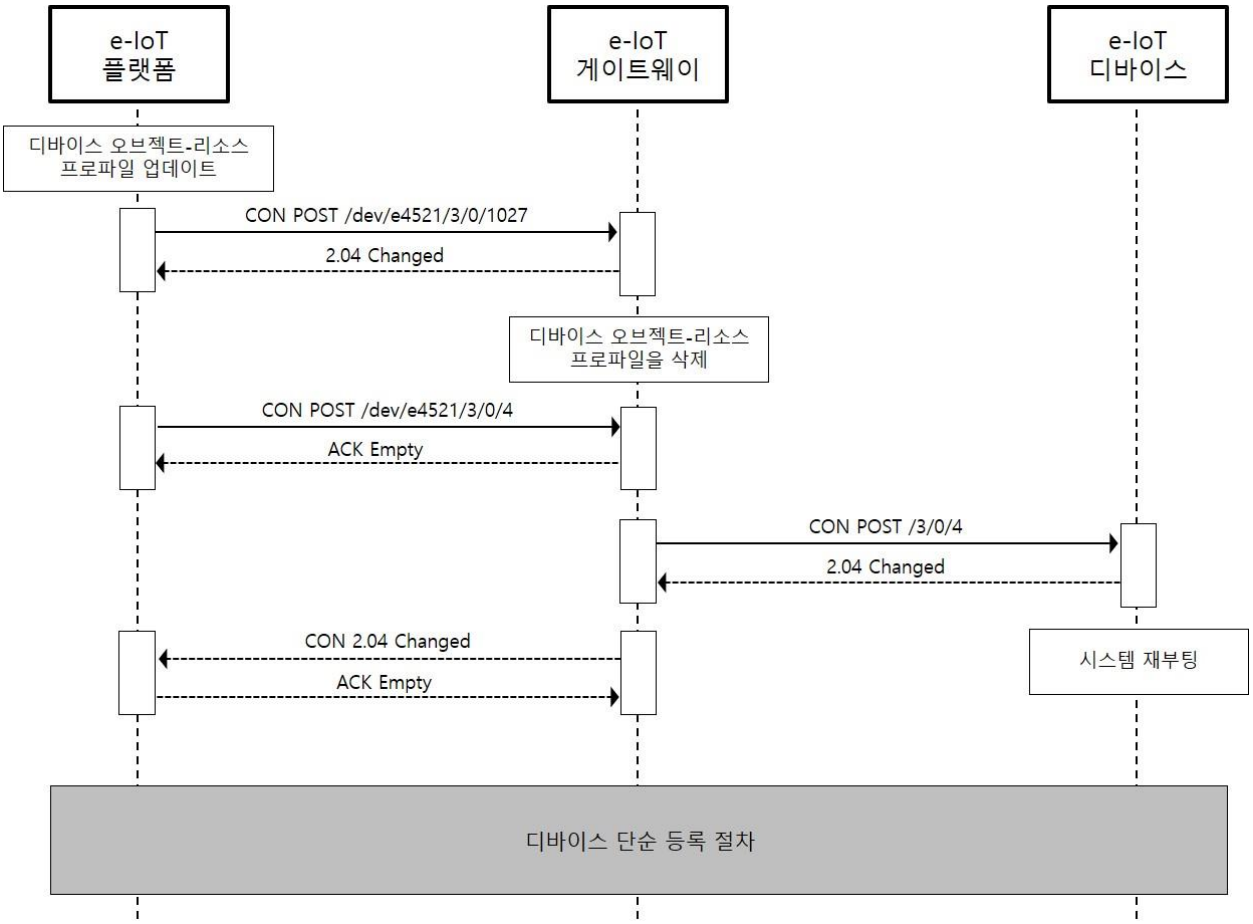

그림 10 — 디바이스 단순등록 업데이트 절차

5.2.3 IFgd- 디바이스 주기적 보고

디바이스의 주기적 보고 관련 기능은 주기적 보고 속성값 설정, CoAP Observe 설정 및 주기보고 및 주기적 보고 해제이다. 각 기능에 대한 디바이스와 게이트웨이의 인터페이스는 **5.1.3** IFpg- 게이트웨이 주기적 보고를 참조한다.

5.2.4 IFgd- 디바이스 단순 주기적 보고

디바이스 단순 주기적 보고는 Device 오브젝트의 Device Type 리소스의 값이 “constrained”인 경우와 해당 리소스의 snot 속성이 True인 경우 수행된다. 이 경우 미리 설정된 pmax 주기로 해당 리소스 값을 주기적으로 전송한다. 주기적 보고 대상이 되는 리소스의 최종 경로는 디바이스 등록 시 획득한 경로(Location-Path) 정보와 사전에 지정한 리소스 경로의 조합으로 결정된다.

디바이스 단순 주기적 보고 인터페이스는 **표 27** 및 **표 28**과 같이 요청 및 응답 메세지를 정의하였고 이 경우 디바이스는 요청자, 게이트웨이는 응답자 역할을 한다.

표 27 — 디바이스 단순 주기적 보고 인터페이스(첫번째 메시지)

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="6">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable 또는
Non-Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="3">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td>location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td colspan="2">Plain-text(0)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td colspan="2">대상 리소스의 값</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.01 Created</td><td rowspan="3">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="2">4.06 Not Acceptable</td></tr>
</table>

표 28 — 디바이스 단순 주기적 보고 인터페이스(두번째 이후 메시지)

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="6">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable 또는
Non-Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">PUT</td></tr>
<tr><td rowspan="3">URI</td><td colspan="2" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td>location</td><td>필수</td><td colspan="2">대상 리소스의 URI</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td colspan="2">Plain-text(0)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td colspan="2">대상 리소스의 값</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="2">4.04 Not Found</td></tr>
</table>

5.2.5 IFgd- 디바이스 정보 조회

디바이스의 정보 조회 인터페이스는 **표 29**와 같이 요청 및 응답 메세지를 정의하였고 이 경우 게이트웨이는 요청자, 디바이스는 응답자 역할을 한다.

표 29 — 디바이스 정보 조회 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	GET
	URI	Template		—	/{location}
					Uri-Path: location
		변수	location	필수	대상 리소스의 URI
	Accept			필수	Plain-text(0) 또는 JSON(11543)
응답 메시지	응답 코드	2.05 Content	Content Format	필수	요청 메시지의 Accept와 동일
			Payload		해당 리소스의 값
		4.04 Not Found			실패

그림 11은 플랫폼에서 디바이스의 정보를 조회하는 전체 과정이다. 플랫폼과 게이트웨이간 정보조회 과정은 본 문서의 **5.1.5** IFpg- 게이트웨이 정보 조회를 참고한다.

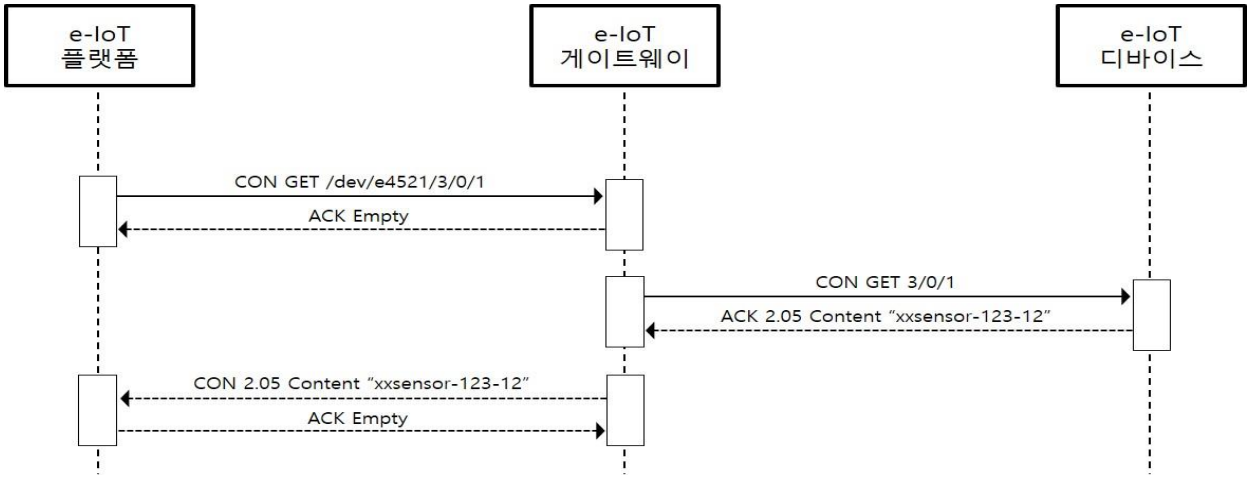

그림 11 — 디바이스 정보 조회 전체 절차

5.2.6 IFgd- 디바이스 제어

디바이스 제어 인터페이스는 **표 30**과 같이 요청 및 응답 메시지로 정의되며 게이트웨이는 요청자, 디바이스는 응답자 역할을 한다.

표 30 — 제어(쓰기 및 실행) 인터페이스

메시지 종류	속성			필수 여부	정의 및 설명
요청 메시지	Transaction Type			필수	Confirmable
	Method Type			필수	PUT(쓰기) 또는 POST(실행)
	URI	Template		—	/{location}
					Uri-Path: location
		변수	location	필수	대상 리소스의 URI
	Content Format			선택	PUT인 경우만 Plain-text(0) 또는 JSON(11543)
	Payload			선택	PUT인 경우만 쓰고자 하는 값
응답 메시지	응답 코드	2.04 Changed		필수	성공
		4.00 Bad Request			실패
		4.04 Not Found			

그림 12는 플랫폼에서 디바이스를 제어하기 위한 전체 과정이다. 플랫폼과 게이트웨이간 정보조회 과정은 본 문서의 **5.1.6** IFpg- 게이트웨이 제어를 참고한다.

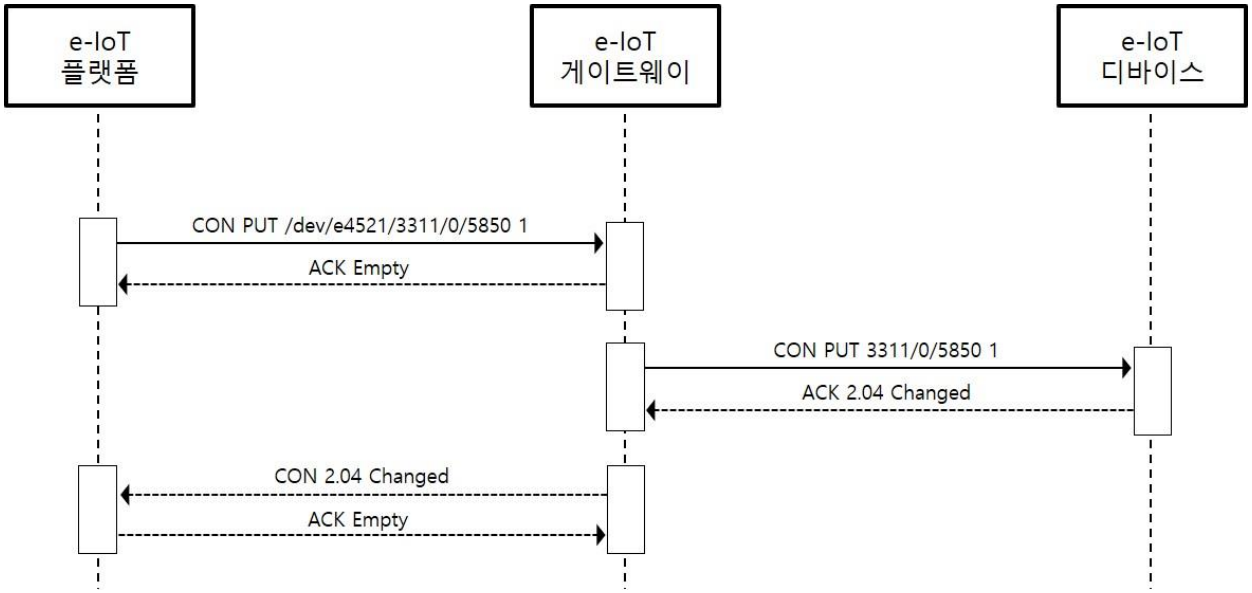

그림 12 — 디바이스 제어 전체 절차

5.3 플랫폼과 디바이스간 인터페이스(IFpd)

플랫폼과 디바이스 사이의 정보 교환을 위한 인터페이스로서, 디바이스 등록, 주기적 보고, 단순 주기적 보고, 정보 조회, 제어 및 펌웨어 업데이트 기능을 위해 사용된다. 특히 등록 과정은 디바이스 기본등록과 단순등록 기능으로 나뉜다.

5.3.1 IFpd- 디바이스 기본 등록

그림 13과 같이 디바이스 기본 등록은 플랫폼에 디바이스의 오브젝트-리소스 등록 수행을 의미한다. 각 오브젝트별 리소스 조회 과정은 선택적으로 수행된다.

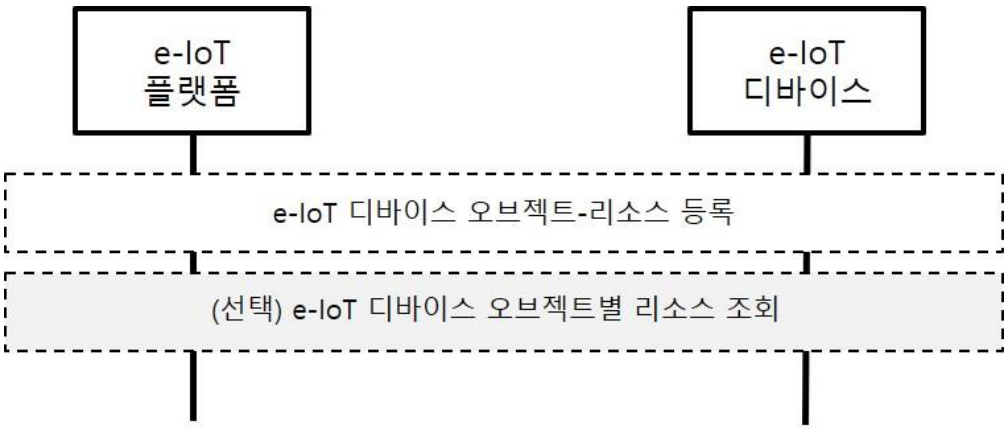

그림 13 — 디바이스 기본 등록 절차

디바이스의 오브젝트-리소스 등록 과정을 수행한 결과 플랫폼에 등록된 e-IoT 디바이스의 데이터 모델의 형상은 **그림 14**와 같다. 디바이스 등록 시 전달되는 경로 정보가 각각 생성되며 그 하부에

오브젝트와 리소스 정보가 저장된다.

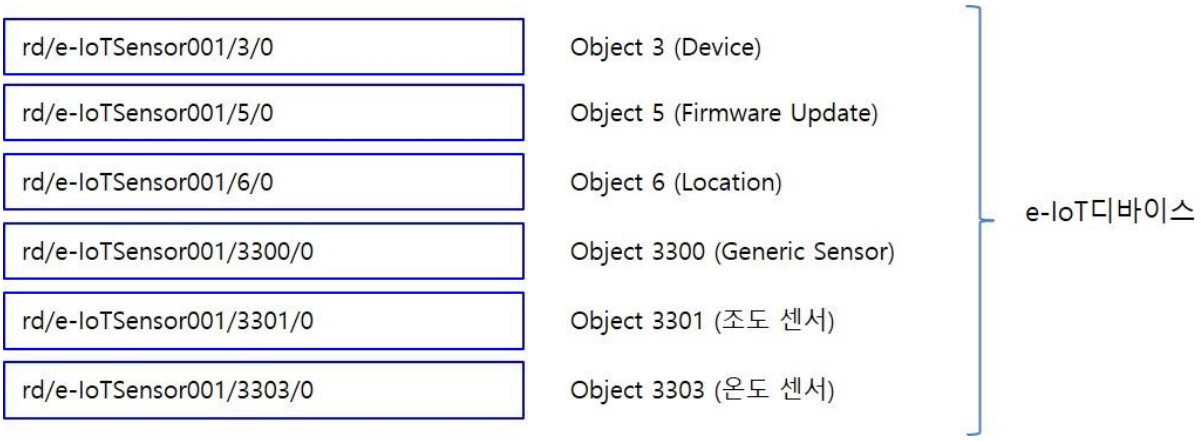

그림 14 — 플랫폼에 등록된 디바이스 데이터모델

5.3.1.1 IFpd: 디바이스 기본 등록

디바이스의 기본 등록 인터페이스는 **표 31**과 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 플랫폼은 응답자 역할을 한다.

표 31 — 디바이스 기본 등록 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="11">요청 메시지</td><td colspan="3">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="3">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="4">URI</td><td rowspan="4">Template</td><td rowspan="4">—</td><td colspan="2">/{+rd}{?ep, lt, CK}</td></tr>
<tr><td>Uri-Path</td><td>rd</td></tr>
<tr><td>Uri-Query</td><td>ep, lt, CK</td></tr>
<tr><td rowspan="4">변수</td><td>rd</td><td>필수</td><td colspan="2">리소스 디렉토리</td></tr>
<tr><td>ep</td><td>필수</td><td colspan="2">Endpoint Name(디바이스 OID)</td></tr>
<tr><td>lt</td><td>필수</td><td colspan="2">Lifetime</td></tr>
<tr><td>CK</td><td>선택</td><td colspan="2">시간 동기 요청</td></tr>
<tr><td colspan="3">Content Format</td><td>필수</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="3">Payload</td><td>필수</td><td colspan="2">디바이스에 정의된 오브젝트와 오브젝트 인스턴스 정보</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="2">2.01 Created</td><td rowspan="3">필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 Location-Path 헤더 포함
— 요청 메시지에 CK 포함 시 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="2">4.03 Forbidden</td></tr>
</table>

5.3.1.2 IFpd: 디바이스 기본 등록 업데이트

디바이스의 기본 등록 업데이트를 위한 인터페이스는 **표 32**와 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 플랫폼은 응답자 역할을 한다.

표 32 — 디바이스 기본 등록 업데이트 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="8">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="4">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{+location}{?lt, CK}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td rowspan="3">변수</td><td colspan="2">Uri-Query</td><td>lt, CK</td></tr>
<tr><td colspan="2">location</td><td>필수</td><td colspan="2">등록과정에서 디바이스가 수신한 경로정보</td></tr>
<tr><td colspan="2">lt</td><td>선택</td><td colspan="2">Lifetime</td></tr>
<tr><td colspan="2">CK</td><td>선택</td><td colspan="2">시간 동기 요청</td></tr>
<tr><td colspan="4">Content Format</td><td>필수</td><td colspan="2">application/link-format(40)</td></tr>
<tr><td colspan="4">Payload</td><td>선택</td><td colspan="2">수정된 리소스 데이터가 있는 경우 포함</td></tr>
<tr><td rowspan="3">응답 메시지</td><td rowspan="3">응답 코드</td><td colspan="3">2.04 Changed</td><td rowspan="3">필수</td><td colspan="2">성공, 요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td colspan="3">4.00 Bad Request</td><td colspan="2" rowspan="2">실패</td></tr>
<tr><td colspan="3">4.04 Not Found</td></tr>
</table>

5.3.1.3 IFpd: 디바이스 기본 등록 해제

디바이스의 기본 등록 해제 인터페이스는 **표 33**과 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 플랫폼은 응답자 역할을 한다.

표 33 — 디바이스 기본 등록 해제 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="4">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="5">요청 메시지</td><td colspan="4">Transaction Type</td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="4">Method Type</td><td>필수</td><td colspan="2">DELETE</td></tr>
<tr><td rowspan="3">URI</td><td colspan="3" rowspan="2">Template</td><td rowspan="2">—</td><td colspan="2">/{location}</td></tr>
<tr><td>Uri-Path</td><td>location</td></tr>
<tr><td>변수</td><td>location</td><td>필수</td><td colspan="2">등록과정에서 디바이스가 수신한 경로정보</td></tr>
<tr><td rowspan="2">응답 메시지</td><td rowspan="2">응답 코드</td><td colspan="3">2.02 Deleted</td><td rowspan="2">필수</td><td colspan="2">성공</td></tr>
<tr><td colspan="3">4.04 Not Found</td><td colspan="2">실패</td></tr>
</table>

5.3.2 IFpd- 디바이스 단순 등록

디바이스 단순 등록은 반드시 사전에 디바이스의 오브젝트-리소스 프로파일 정보가 플랫폼에 저장되어 있어야 하며, 등록 요청 메시지에 lep 옵션을 포함해야 한다.

5.3.2.1 IFpd: 디바이스 단순 등록

디바이스 단순 등록을 위한 인터페이스는 **표 34**와 같이 요청 및 응답 메시지로 정의되며 디바이스는 요청자, 플랫폼은 응답자 역할을 한다.

표 34 — 디바이스 단순 등록 인터페이스

<table>
<tr><th>메시지 종류</th><th colspan="3">속성</th><th>필수 여부</th><th colspan="2">정의 및 설명</th></tr>
<tr><td rowspan="9">요청 메시지</td><td colspan="2">Transaction Type</td><td></td><td>필수</td><td colspan="2">Confirmable</td></tr>
<tr><td colspan="2">Method Type</td><td></td><td>필수</td><td colspan="2">POST</td></tr>
<tr><td rowspan="6">URI</td><td rowspan="3" colspan="2">Template</td><td rowspan="3">—</td><td colspan="2">/{+rd}{?lt, lep, CK}</td></tr>
<tr><td>Uri-Path</td><td>rd</td></tr>
<tr><td>Uri-Query</td><td>lt, lep, CK</td></tr>
<tr><td rowspan="4">변수</td><td>rd</td><td>필수</td><td colspan="2">리소스 디렉토리</td></tr>
<tr><td>lt</td><td>필수</td><td colspan="2">Lifetime</td></tr>
<tr><td>lep</td><td>필수</td><td colspan="2">Lightweight Endpoint Name(OID)</td></tr>
<tr><td>CK</td><td>선택</td><td colspan="2">시간 동시 요청</td></tr>
<tr><td rowspan="4">응답 메시지</td><td rowspan="4">응답 코드</td><td colspan="2">2.01 Created</td><td rowspan="4">필수</td><td colspan="2">— 등록된 오브젝트-리소스 정보에 해당하는 경로 정보(Location-Path) 헤더 포함
— 요청 메시지에 CK 포함 시, 페이로드에 플랫폼의 시스템 시간 포함</td></tr>
<tr><td colspan="2">4.00 Bad Request</td><td colspan="2" rowspan="3">실패</td></tr>
<tr><td colspan="2">4.03 Forbidden</td></tr>
<tr><td colspan="2">4.04 Not Found</td></tr>
</table>

5.3.2.2 IFpd: 디바이스 단순 등록 업데이트

디바이스 단순 등록 업데이트 절차는 다음과 같다.

- 플랫폼에서 수정된 디바이스 오브젝트-리소스 프로파일이 존재할 경우 플랫폼이 디바이스에 해당 디바이스의 오브젝트-리소스 프로파일 삭제 및 재부팅 요청 메시지를 보낸다.
- 디바이스는 시스템 재부팅 후 **5.3.2.1** IFpd: 디바이스 단순 등록을 수행한다.

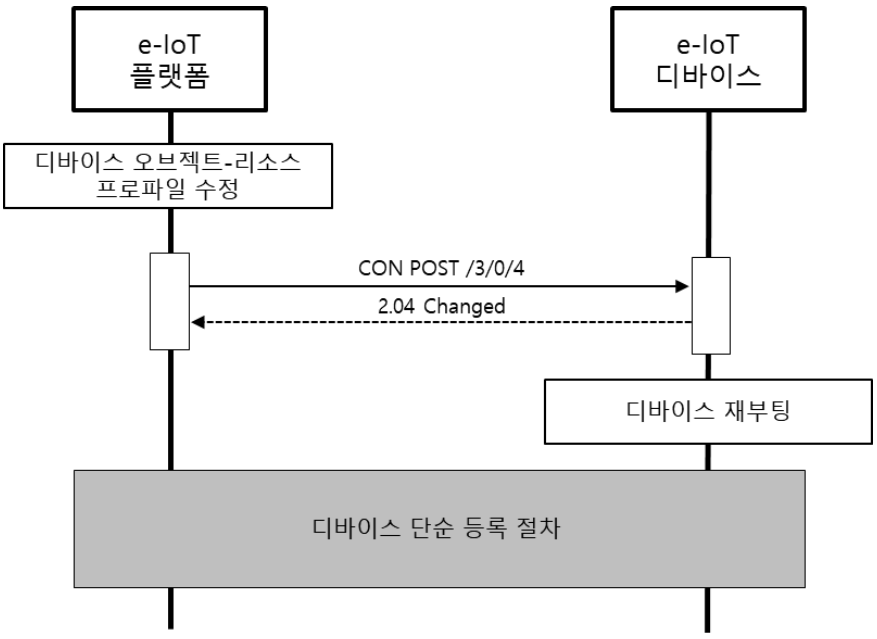

그림 15 — 디바이스 단순 등록 업데이트 절차

5.3.3 IFpd- 디바이스 주기적 보고

디바이스의 주기적 보고 관련 기능은 주기적 보고 속성값 설정, CoAP Observe 설정 및 주기적 보고, 주기적 보고 해제이다. 각 기능에 대한 디바이스와 플랫폼의 인터페이스는 **5.1.3** IFpg - 게이트웨이 주기적 보고를 참조한다.

5.3.4 IFpd- 디바이스 단순 주기적 보고

디바이스와 플랫폼 사이의 네트워크 대역폭이 작은 열악한 환경에서 단순 주기적 보고 기능이 활용된다. 디바이스와 플랫폼 간 단순 주기적 보고 인터페이스는 **5.1.4** IFpg - 게이트웨이 단순 주기적 보고를 참고한다.

5.3.5 IFpd- 디바이스 정보 조회

플랫폼이 디바이스의 정보를 조회하는 기능은 해당 오브젝트와 리소스의 속성값을 읽어오는 Read 동작과 해당 오브젝트에 구현된 리소스를 조회하는 Discovery 동작으로 구분된다. 디바이스와 플랫폼 간 디바이스 정보 조회 인터페이스는 **5.1.5** IFpg - 게이트웨이 정보 조회를 참고한다.

5.3.6 IFpd- 디바이스 제어

플랫폼이 디바이스를 제어하는 기능은 대상 리소스에 값을 설정하여 쓰기(Write) 및 실행(Execute) 동작을 목적으로 한다. 디바이스와 플랫폼 간 디바이스 제어 인터페이스는 **5.1.6** IFpg - 게이트웨이 제어를 참고한다.

5.3.7 IFpd- 디바이스 펌웨어 업데이트

플랫폼과 디바이스 간 펌웨어 업데이트 절차는 펌웨어 파일 전달, 다운로드 상태 확인, 업데이트 수행(디바이스 재부팅), 디바이스 재등록 및 업데이트 결과 확인 과정으로 진행된다. 디바이스의 펌웨어 업데이트 기능은 하드웨어 상황에 따라 선택적으로 구현될 될 수 있으며, 디바이스와 플랫폼 간 디바이스 펌웨어 업데이트 인터페이스는 **5.1.7** IFpg - 게이트웨이 펌웨어 업데이트를 참고한다.

6 e-IoT 정보모델링

6.1 개요

e-IoT 정보모델링은 LwM2M 정보모델링을 기반으로 하며, 정보모델 구조는 다음과 같이 4개의 계층적 구조를 갖는다.

- {Object-id}/{Object Instance}/{Resource-id}/{Resource Instance}

그림 16은 게이트웨이의 정보모델링 구조로 게이트웨이 자체 모델링과 게이트웨이에 연결된 디바이스의 정보모델을 포함한 구조를 갖는다. 게이트웨이에 연결된 디바이스의 정보모델 또한 4개의 계층적 구조를 가지며 경로 정보는 “dev/e{장치 등록 시 게이트웨이에서 할당한 임의 번호}”와 같다.

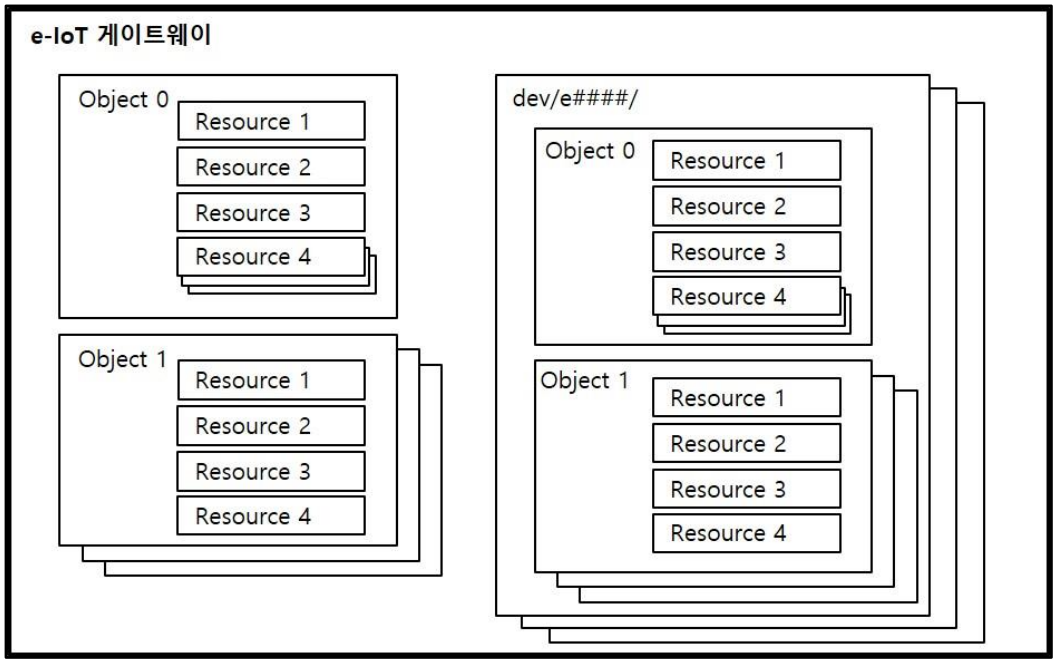

그림 16 — e-IoT 게이트웨이 정보모델링 구조

6.1.1 오브젝트와 리소스 ID 범위

e-IoT 정보모델링의 오브젝트 및 리소스 범위는 **표 35** 및 **표 36**과 같다.

표 35 — e-IoT 오브젝트 범위

분류	e-IoT 오브젝트 ID 범위	설명
OMA 오브젝트	0 - 1023	OMA에서 정의한 오브젝트 (e-IoT 정보모델링에 활용)
미래 사용 오브젝트	1024 - 2047	미래 사용을 위한 예약된 오브젝트
외부표준기관 사용 오브젝트	2048 - 10240	제3의 표준 단체에서 사용할 목적의 오브젝트
사설 사용 오브젝트	10241 - 26240	개인 또는 기업이 OMA에 등록하여 사용 할 수 있는 오브젝트 (e-IoT 정보모델링에 활용)
	26241 - 32768	사업자에서 사용할 목적의 오브젝트, OMA 등록 안함 (e-IoT 정보모델링에 활용)
	32769 - 42768	사업자들이 오브젝트 블록 단위로 등록 사용(최대 50개), 등록된 블록에 속하는 오브젝트는 자체 이용 가능

표 36 — e-IoT 리소스 범위

분류	e-IoT 리소스 ID 범위	설명
공통 리소스	0 - 2047	LwM2M 규격에 정의된 해당 오브젝트별 리소스 (e-IoT 정보모델링에 활용)
재사용 가능 리소스	2048 - 26240	모든 오브젝트에 대해서 재사용 가능한 리소스 (e-IoT 정보모델링에 활용)
사설 이용 리소스	26241 - 32768	사적으로 사용 가능, OMA 등록 안함 (e-IoT 정보모델링에 활용)

6.1.2 오브젝트와 리소스 속성

데이터 보고 및 통보와 관련된 e-IoT 정보모델링(오브젝트와 오브젝트 인스턴스, 리소스와 리소스 인스턴스) 속성값은 **표 37**과 같다. snot 속성은 단순 주기적 보고 기능 수행을 위해 사용되며, 플랫폼제어를 통해 속성값 변경이 가능하다.

표 37 — 데이터 보고/통보 관련 LwM2M 속성(선택)

속성 이름	코어링크 변수 (Core Link Param)	속성 적용 위치 (Attachment)	속성 명시 위치 (Assignation Level)	데이터 타입
Minimum Period	pmin	Resource	Resource Level Object Level Object Instance Level	Integer
Maximum Period	pmax	Resource	Resource Level Object Level Object Instance Level	Integer
Greater Than	gt	Resource	Resource Level	Float
Less Than	lt	Resource	Resource Level	Float
Step	st	Resource	Resource Level	Float
Simple Notification	snot	Resource	Resource Level	Boolean

디바이스에서 Historical 데이터를 게이트웨이와 플랫폼에 전송하기 위해서는 JSON 형식의 Historical 데이터 표현을 위해 e-IoT 정보모델 속성값을 **표 38**과 같이 정의한다. 3개의 속성으로 정의되며, Historical 데이터 표현 지원 여부를 설정하는 데 사용된다. Hrep 속성은 해당 리소스가 Historical 데이터 지원 여부 결정에 사용되고, Hmax 속성은 저장되는 Historical 데이터 값의 개수를 정의한다. 기본값이 5로 되어 있기 때문에 별도 설정이 없을 경우 5개의 과거 값이 저장된다. 마지막으로 Hrst 속성은 저장된 과거 값들을 제거할 때 사용한다.

표 38 — Historical 데이터 지원 LwM2M 속성

속성 이름	코어링크 변수 (Core Link Param)	속성 적용 위치 (Attachment)	속성 명시 위치 (Assignation Level)	데이터 타입
Historical Representation	Hrep	Resource	Resource Level	Boolean, (기본값: false)
Maximum number of historical values	Hmax	Resource	Resource Level	Integer, (기본값: 5)
Reset historical values	Hrst	Resource	Resource Level	Boolean, (기본값: false)

6.1.3 데이터 타입과 데이터 포맷

e-IoT 오브젝트-리소스의 데이터 타입은 **표 39**와 같이 정의한다.

표 39 — e-IoT 오브젝트-리소스 데이터 타입

데이터 타입	설명
String	UTF-8 String
Integer	8, 16, 32, 64-bit Signed Integer
Float	32 또는 64-bit Floating Point Value.
Boolean	8-bit Unsigned Integer (0: False, 1: True)
Opaque	Binary Octets
Time	Unix Time으로 UTC time zone의 1970년 1월 1일 이후의 시간을 초로 표현 (signed integer)
Objlnk	Object Link로 Object Instance를 참조하는 데 사용(두 개의 16bit unsigned integer로 구성되며 첫 번째 값은 Object ID, 두 번째 값은 Object Instance ID임)
Rsclnk	Resource Link로 Resource나 Resource Instance를 참조하는데 사용(string 문자열로 표현하며 오브젝트-리소스 표현 방식을 따름)

e-IoT의 리소스 전달을 위한 데이터 포맷은 다음과 같이 Plain Text, Core Link Param, Opaque, JSON, TLV를 지원한다.

- Plain Text(text/plain; charset=utf-8): 0
- Core Link Param(application/link-format): 40
- Opaque(application/octet-stream): 42
- JSON(application/vnd.oma.LwM2M+json): 11543
- TLV(application/vnd.oma.LwM2M+tlv): 11542

e-IoT 리소스의 JSON 데이터 포맷의 변수는 **표** 40와 같다.

표 40 — e-IoT JSON 변수 정의

속성	JSON 변수		설명
Base Name	bn		모든 리소스의 루트가 되는 위치
Base Time	bt		기본이 되는 현재 시간
Resource Array	Array Parameters	e	리소스 리스트
	Name	n	리소스 인스턴스의 경로 이름
	Time	t	초 단위, bt를 기준으로 상대적인 시간 값
	Float Value	v	Integer, Float, Time인 경우
	Boolean Value	bv	JSON Boolean (0, 1, True, False 가능)
	Object Link Value	ov	리소스 데이터 타입이 Objlnk인 경우
	String Value	sv	Boolean, Integer, Float, Time 이외의 경우
	Extra Type	ex	Executable, Opaque type의 리소스를 표현, 게이트웨이와 디바이스의 OBJ-RSC 프로파일에만 사용되며 값은 0으로 표현
	Resource Link Value	rv	리소스 데이터 타입이 Rsclnk인 경우

6.2 복합 데이터

2종 이상의 센서를 탑재하고 있는 e-IoT 장치의 경우 보고할 센서의 측정값이 다수 존재한다. 이 경우 주기적 보고는 각 센서별로 CoAP Observe 속성 설정을 수행하여 주기적 보고를 수행해야 한다. 이러한 주기적 보고 방식은 네트워크 대역폭이 작은 장치에게는 한계가 있어 다수의 센서 측정값을 하나의 보고 메시지에 전달할 수 있는 복합 데이터 보고 기능이 효율적이다.

그림 17과 같이 게이트웨이의 온도센서, 게이트웨이에 연결된 두 개의 디바이스의 전압센서와 전류센서의 측정값들이 주기적으로 보고 될 수 있다. 즉, 물리적으로 분산되어 존재하는 여러 센서의 측정 데이터들을 가상의 복합 센서를 정의하여 보고한다.

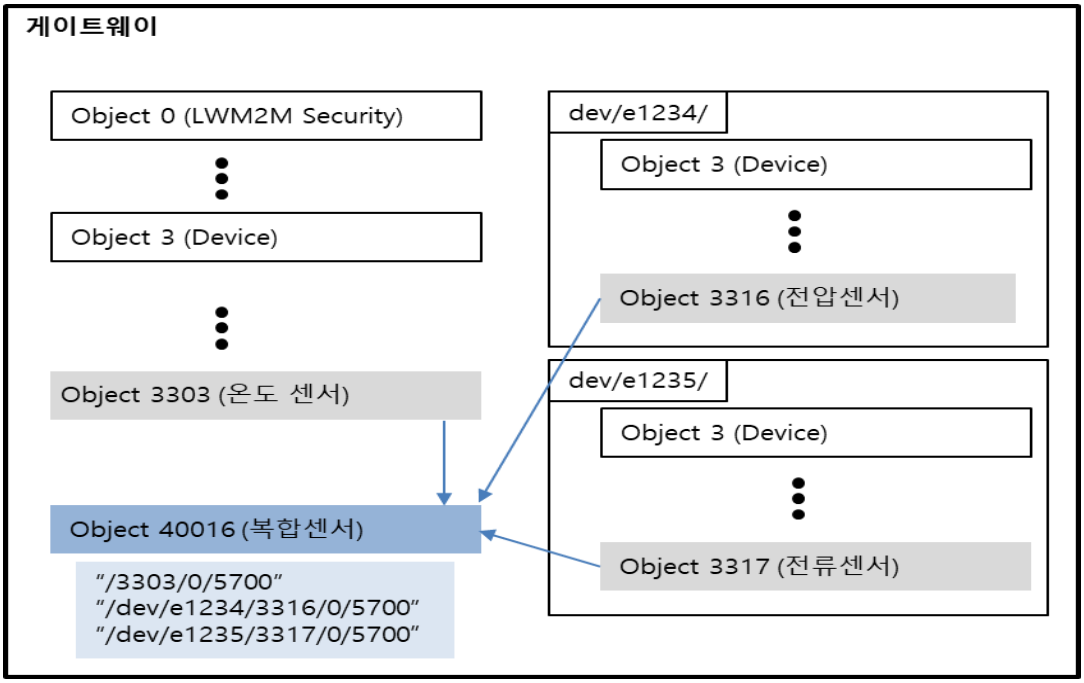

그림 17 — 복합 센서 오브젝트를 포함한 게이트웨이 정보 모델

6.2.1 복합 센서 오브젝트와 리소스 정의

복합 센서 오브젝트는 **표 41**과 같이 정의한다.

표 41 — 복합 센서 오브젝트 정의

이름	오브젝트 ID	오브젝트 인스턴스	필수/선택	URN
복합 센서	40016	Multiple	선택	urn:oma:lwm2m:oma:40016

복합 센서 오브젝트의 리소스는 **표 42**와 같이 정의한다.

표 42 — 복합 센서 오브젝트의 리소스 정의

ID	이름	동작	리소스 인스턴스	선택/필수	타입	설명
0	복합 센서 값	R,W	Single	필수	String	복합 센서를 구성하는 센서들의 측정값으로 ‘;’로 구분
1	복합 센서 구성	R,W	Single	필수	Rsclnk	복합 센서를 구성하는 센서 값의 URL로 ‘;’로 구분
2	복합 센서 값 단위	R,W	Single	선택	String	복합 센서 값의 각 단위로 ‘;’로 구분
3	복합 센서 값 개수	R,W	Single	선택	Integer	복합 센서를 구성한 센서 값의 개수

그림 17의 장치의 복합 센서의 리소스 값들은 다음과 같다.

표 43 — 복합 센서 리소스 값

복합 센서 리소스	값
/40016/0/0	37;220;60
/40016/0/1	/3303/0/5700;/dev/e1234/3316/0/5700;/dev/e1235/3317/0/5700
/40016/0/2	Celsius;volt;ampere
/40016/0/3	3

6.3 Historical 데이터

네트워크 대역폭이 작은 디바이스는 측정된 데이터를 매번 전송하지 않고 일정 시간 모아서 게이트웨이 및 플랫폼에 전송하는 것이 효율적이다. 이와 같이 현재 측정된 값과 과거에 측정된 값을 포함하는 데이터를 Historical 데이터라고 한다. Historical 데이터는 JSON의 시간(Time) 변수를 이용하는 방법과 Historical 데이터를 지원하는 리소스를 통해 표현된다.

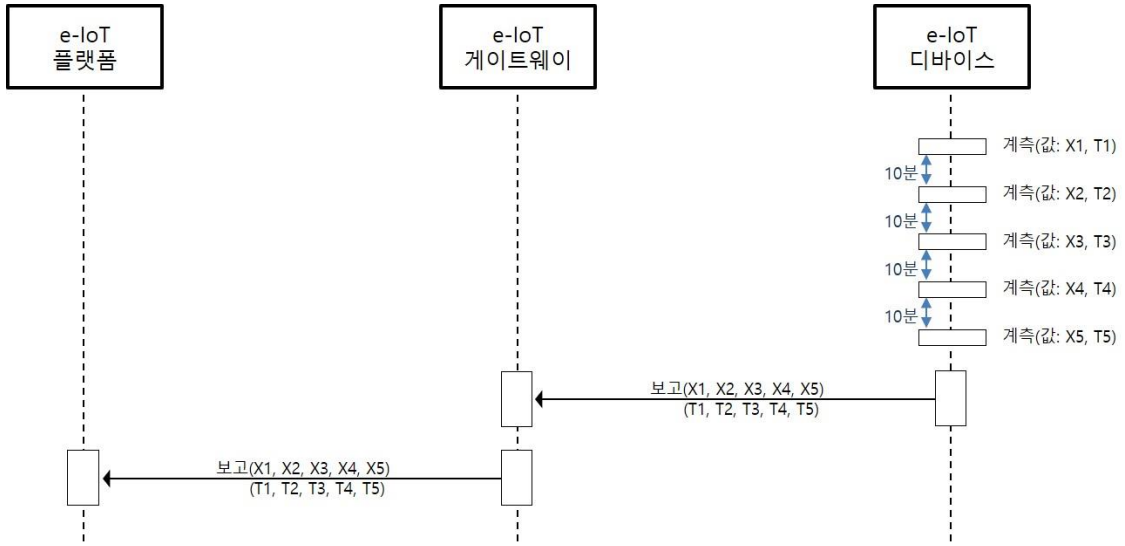

그림 18 — Historical 데이터 보고 개요

6.3.1 JSON 시간(Time) 변수 기반 Historical 데이터 보고

JSON의 시간(Time) 변수를 활용한 Historical 데이터 메시지 형식 및 예시는 **표 44**와 같다.

표 44 — JSON 시간변수 기반 Historical 데이터 메시지 형식 및 예시

속성	JSON 변수		설명	메시지 예시
Base Name	bn		모든 리소스의 루트가 되는 위치	{ “bn”:”/dev/e1234/3316/0/5700” “e”: [{“n”: “”, “v”: “220.1”, “t”:-5}, {“n”: “”, “v”: “220.0”, “t”:-30}] “bt”: 25662634 }
Base Time	bt		기본이 되는 현재 시간	
Resource Array	Array Parameters	e	리소스 리스트	
	Name	n	리소스 인스턴스의 경로 이름	
	Time	t	초 단위, bt를 기준으로 상대적인 시간 값	
	Float Value	v	Integer, Float, Time인 경우	

6.3.2 Historical 데이터 지원 Resource 기반

Historical 데이터는 **표 45**와 같이 Historical 데이터를 지원하는 추가 리소스를 이용하여 표현할 수 있다.

표 45 — Historical 데이터 관련 리소스

ID	이름	동작	인스턴스	필수/선택	타입	범위	단위	설명
30058	Historical Sensor Value	R,W	Multiple	선택	Float	-	별도	Max Historical Instance 리소스의 값까지 인스턴스를 생성해서 값을 저장하며 최대값까지 저장된 경우 첫 Instance부터 다시 값을 덮어씀
30059	Stored Time	R,W	Multiple	선택	Time	-	-	Historic Sensor Value 리소스가 저장되는 시간으로 동일 Resource Instance ID 사용
30060	Max Historical Instance	R,W	Single	선택	Integer	-	-	Historical Sensor Value가 허용되는 Instance 개수
30061	Reset Historical Values	E	Single	선택	-	-	-	실행될 경우 Historical Sensor Value의 모든 Instance를 삭제함
30062	Historical Resource Link	R,W	Multiple	선택	Rsclnk	-	-	Historical Sensor Value와 Stored Time 리소스를 링크

6.4 DLMS 데이터 정보모델링

6.4.1 DLMS 자원관리 구조

다수의 전력량계는 IEC 62056(DLMS/COSEM) 표준을 준용하고 있으며 전력량계의 데이터를 e-IoT 플랫폼에서 수집 및 제어하기 위해서는 DLMS 데이터 정보모델링 정의가 필요하다. DLMS 기반 전력량계는 보유할 수 있는 계량데이터를 포함한 모든 데이터에 대해서 객체 모델링 방식을 지원한다.

DLMS 객체 모델링 시 데이터에 부여하는 고유한 이름이 OBIS(Object Identification System) 코드이며, **그림 19**와 같이 OBIS 코드는 6 바이트 크기를 가지며, 계층적 구조를 갖는 것이 특징이다. OBIS 코드의 그룹 A은 에너지 관련 분야인 전기, 가스, 수도, 열량 등을 구분하고 그룹 B는 측정 채널 또는 통신 채널을 구분하는데 사용한다. 그룹 C는 그룹 A와 연관하여 전류, 전압, 전력, 부피, 온도 등과 같은 물리적 데이터를 구분한다. 그룹 D와 그룹 E는 각각 그룹 A와 그룹 C로 분류 된 데이터의 추가 연산과 세분화를 위해 사용된다. 그룹 F는 데이터에 시간을 부여하기 위해서 사용하며, 현월 검침 값, 전월 검침 값 등을 표현한다.

OBIS 그룹A (1byte), 0~15, Medium	OBIS 그룹B (1바이트, 0~64, Channel	OBIS 그룹C (1바이트), 0~255, Quantity	OBIS 그룹D (1바이트), 0~255, Processing	OBIS 그룹E (1바이트), 0~255, Classification	OBIS 그룹F (1바이트), 0~255 Historical value
1 Electricity	1 Channel	1 ∑LiA+	8 Time Integral	2 Rate	255 Current

그림 19 — DLMS에서 사용하는 OBIS 코드 구조

6.4.2 DLMS 데이터 정보모델링 구조

6.4.2.1 연동 구조 및 요구사항

그림 20은 e-IoT 시스템에서 DLMS 정보모델링 기반의 데이터 교환을 위한 연동 구조이다. e-IoT 게이트웨이 또는 디바이스는 DLMS 기기인 전력량계로부터 계량 데이터를 수집하여 e-IoT의 DLMS 정보모델링 기반의 변환을 통해 e-IoT 플랫폼에 데이터를 전달한다. 또한, 게이트웨이는 플랫폼으로부터 설정 값 변경 등의 요청 메시지에 대해서 DLMS 정보모델링 기반의 자원관리 구조를 생성하여 동작한다.

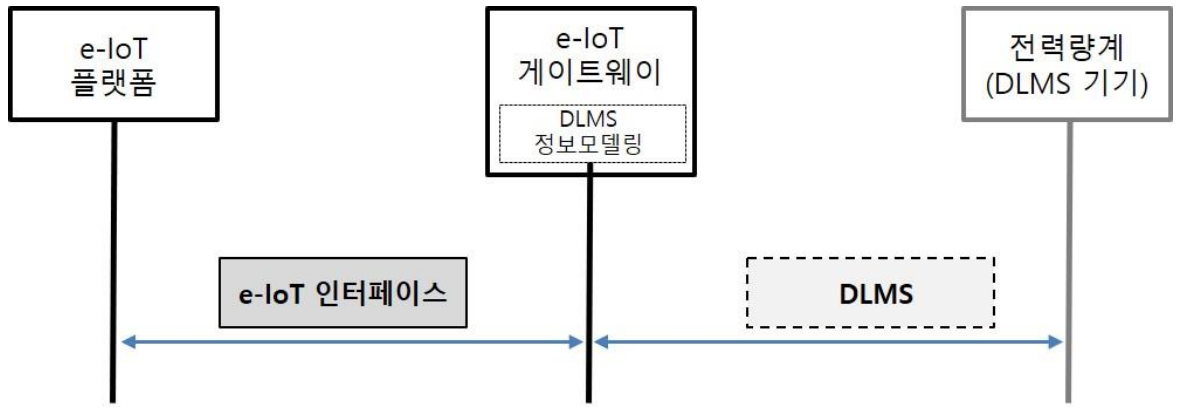

그림 20 — e-IoT의 DLMS 정보모델링 연동 구조

e-IoT의 DLMS의 정보모델링 요구사항을 다음과 같이 요약할 수 있다.

- e-IoT 오브젝트-리소스 모델과 DLMS 객체 모델링은 1:1로 매핑 필요
- 매핑 완료된 정보모델링 결과는 e-IoT 인터페이스 및 DLMS 표준 규격에 부합

6.4.2.2 e-IoT DLMS 정보모델링 방법

그림 21과 **그림 22**는 e-IoT 게이트웨이의 DLMS 정보모델링을 위한 DLMS 객체 모델링을 e-IoT 오브젝트-리소스 모델로 매핑하는 상호 연동 방식을 보여준다.

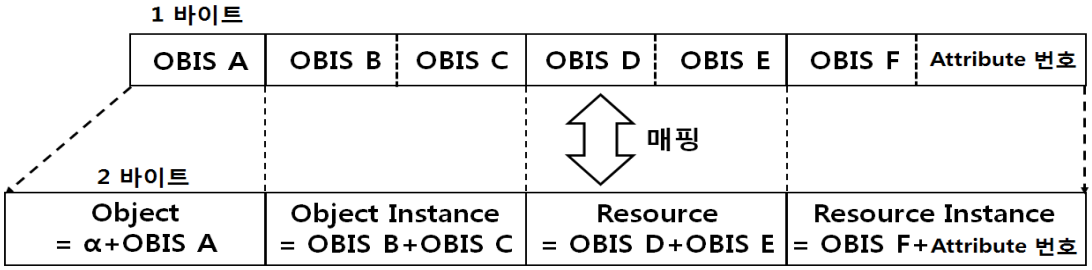

그림 21 — e-IoT의 DLMS 정보모델링 연동 방식(GET/PUT)

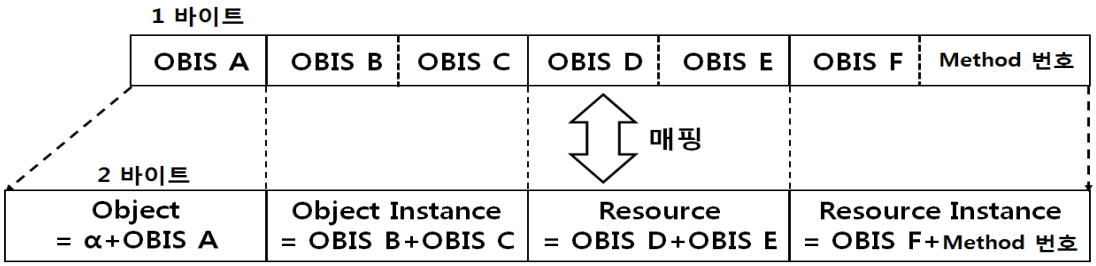

그림 22 — e-IoT의 DLMS 정보모델링 연동 방식(POST)

첫번째 항목인 Object ID는 DLMS의 기본 Object 값으로 정의된 α 값에 OBIS A 값의 합으로 표현한다. DLMS 자원을 표현하는 α 값을 정의하고, DLMS OBIS A 값을 기반으로 Object ID의 범위를 등록한다. 본 규격에서는 **표 46**과 같이 임의 α ~ α +15로 표현한다.

표 46 — e-IoT의 DLMS 정보모델 오브젝트(α=40000 가정 시)

이름	오브젝트 ID	오브젝트 인스턴스	URN
DLMS 자원	34130 - 34145	Multiple	urn:oma:lwm2m:x: 34130 - 34145

두번째 항목인 Object Instance ID는 DLMS의 OBIS B 값에 OBIS C 값의 합으로 매칭한다. 세번째 항목인 Resource ID는 DLMS의 OBIS D 값에 OBIS E 값의 합으로 매칭한다. 마지막 항목인 Resource Instance ID는 DLMS의 OBIS F 값에 서비스 구분에 따라서 Attribute 번호 또는 Method 번호의 합으로 매칭한다.

표 47은 DLMS 오브젝트-리소스 정보모델의 속성을 표현하기 위해 새롭게 추가 정의한 e-IoT의 속성 정보이다.

표 47 — DLMS 오브젝트-리소스 정보모델 관련 속성 정보

속성 이름	코어링크 변수 (Core Link Param)	속성 적용 위치 (Attachment)	속성 명시 위치 (Assignation Level)	데이터 타입
DLMS Attribute	att	Resource Instance	Resource Instance Level	Integer
DLMS Method	met	Resource Instance	Resource Instance Level	Integer

표 48은 DLMS에서 사용하고 있는 날짜 및 시간 항목과 순시 전류 항목의 OBIS 코드를 [(OBIS: 0.0.1.0.0.255, Attribute 번호: 2) 및 (OBIS: 1.0.31.7.0.255, Attribute 번호: 2)] 본 변환 규격을 통해 “/40000/1/0/65282”, “/40001/31/1791/65282” 형태의 e-IoT URI로 각각 표현된다.

표 48 — DLMS 자원관리와 e-IoT 오브젝트-리소스 정보모델 연동 규격

DLMS 자원관리							
OBIS 코드(A~F)						Attribute	의미
A	B	C	D	E	F	번호	—
0	0	1	0	0	255	1	날짜 및 시간 OBIS 코드
0	0	1	0	0	255	2	날짜 및 시간
1	0	31	7	0	255	1	순시전류 OBIS 코드
1	0	31	7	0	255	2	순시 전류값
⇳							
e-IoT의 DLMS 오브젝트-리소스 정보모델(α = 34130 가정)							
Object ID	Object Instance ID		Resource ID		Resource Instance ID		최종 URI
34130 + 0 =34130	256 × 0 + 1 =1		256 × 0 + 0 =0		256 × 255 + 1 =65281		/34130/1/0/65281
34130 + 0 =34130	256 × 0 + 1 =1		256 × 0 + 0 =0		256 × 255 + 2 =65282		/34130/1/0/65282
34130 + 1 =34131	256 × 0 + 31 =31		256 × 7 + 0 =1792		256 × 255 + 1 =65281		/34131/31/1792/65281
34130 + 1 =34131	256 × 0 + 31 =31		256 × 7 + 0 =1792		256 × 255 + 1 =65282		/34131/31/1792/65282

6.5 수요반응 데이터 정보모델링

6.5.1 수요반응 개요

수요반응 서비스는 전기사용자가 전력시장 가격이 높을 때 또는 전력계통 위기 시 아낀 전기를 전력시장에 판매하고 금전으로 보상받는 제도이다. e-IoT 기반 수요반응 데이터 정보모델링을 통해 수요반응 서비스 사업자는 고객에게 수요자원 감축 이벤트를 전달하고, 고객은 내부기기에 제어 명령을 전달하여 수요자원을 감축한다. 본 절에서는 수요반응 서비스를 위한 수요반응 및 미터링 데이터 정보모델링을 정의한다.

6.5.2 수요반응 오브젝트와 리소스 정의

수요반응 오브젝트는 **표 49**와 같이 정의한다.

표 49 — 수요반응 오브젝트 정의

이름	오브젝트 ID	오브젝트 인스턴스	필수/선택	URN
수요반응 (DRLC)	40017	Multiple	선택	urn:oma:lwm2m:x:40017

수요반응 오브젝트의 리소스는 **표 50**과 같이 정의한다.

표 50 — 수요반응 오브젝트의 리소스 정의

ID	이름	동작	리소스 인스턴스	선택/ 필수	타입	단위	설명
0	장치 ID	R	Single	필수	String	-	장치 식별자
1	프로토콜	R	Single	필수	String	-	장치에서 사용하는 프로토콜 (0=LwM2M, 1= OCF, 2= CHIP, 3~= 예약됨)
2	수요반응 ID	R, W	Single	필수	String	-	수요반응 이벤트 식별자
3	수요반응 수준	R, W	Single	선택	Integer	-	4 단계(0~3)로 표현되는 감축 수준, 각 단계별 수준 값은 리소스 4번에서 정의함 “수요반응 감축비율”, “수요반응 감축량”이 없을 경우 필수
4	수요반응 수준 값	R,W	Single	선택	String	-	“수요반응 수준”의 각 단계별 값을 할당하고 구분자로 ‘;’ 를 사용, 수요반응 수준이 정의된 경우 필수 예: “0;30;50;70” 일 경우 “수요반응 수준” 0, 1, 2, 4의 값은 각각 0%, 30%, 50%, 70%를 의미함
5	수요반응 감축비율	R,W	Single	선택	Integer	%	최대소비전력대비 감축해야 할 비율, 수요반응 수준 및 수요반응 감축량이 없을 경우 필수, (예, 30인 경우 최대 소비전력 대비 30% 전력 감축)
6	수요반응 감축량	R, W	Single	선택	Integer	W	최대소비전력대비 감축해야 할 전력량, 수요반응 수준 및 수요반응 감축비율이 없을 경우 필수, (예, 30인 경우 최대 소비전력 대비 30W 전력 감축)
7	생성시간	R, W	Single	필수	Integer	-	수요반응 이벤트 생성 시간
8	시작시간	R, W	Single	필수	Integer	-	수요반응 이벤트 시작 시간
9	지속시간	R, W	Single	필수	Integer	초	수요반응 이벤트 지속되는 시간
10	시작시간 랜덤값	R, W	Single	선택	Float	초	수요반응 이벤트 동시 시작을 방지하기 위한 랜덤 값, (예, −300일 경우 Start Time값의 300초 이내에서 수요반응 이벤트 수행)
11	수요반응 상태	R, W	Single	필수	Integer	-	수요반응 이벤트 처리 상태 (0 = 수요반응 이벤트 수신, 1 = 수요반응 이벤트 시작, 2 = 수요반응 이벤트 완료, 3 = 서버에 의한 수요반응 이벤트 중간 취소, 4 = 사용자 또는 가전 자체 원인에 의한 수요반응 이벤트 취소)

6.5.3 미터링 오브젝트와 리소스 정의

수요반응 서비스를 위한 미터링 오브젝트는 **표 51**과 같이 정의한다.

표 51 — 미터링 오브젝트 정의

이름	오브젝트 ID	오브젝트 인스턴스	필수/선택	URN
미터링	40018	Multiple	선택	urn:oma:lwm2m:x:40018

미터링 오브젝트의 리소스는 **표 52**와 같이 정의한다.

표 52 — 미터링 오브젝트의 리소스 정의

ID	이름	동작	리소스 인스턴스	선택/필수	타입	단위	설명
0	장치 ID	R	Single	필수	String	-	장치 식별자
1	프로토콜	R	Single	선택	String	-	장치에서 사용하는 프로토콜 (0 = LwM2M, 1 = OCF, 2 = CHIP, 3~ = 예약됨)
2	미터링 종류	R	Single	선택	Integer	-	미터링 종류 (0 = 전기, 1 = 가스, 2 = 수도, 3 = 시간, 4 = 압력, 5 = 열, 6 = 냉각, 7~ = 예약)
3	장치 상태	R	Single	필수	Boolean	-	장치의 현재 상태, (False = off, True = on)
4	미터링 보고 형태	R	Single	필수	Integer	-	보고되는 미터링값이 시간 구간인지, 순시값인지 표현 (1 = 구간값, 2 = 순시값)
5	시작시간	R,W	Single	선택	Integer	-	미터링 시작 시간 미터링 보고 형태가 구간값(1)인 경우 필수
6	측정기간	R,W	Single	선택	Integer	-	미터링 측정 기간 미터링 보고 형태가 구간값(1)인 경우 필수
7	미터링 값	R	Single	선택	Float	-	미터링 값, 미터링 보고 형태가 순시값(2)인 경우 필수
8	미터링 배열 값	R	Single	선택	Array	-	미터리 값의 배열, 미터링 보고 형태가 구간값(1)인 경우 필수
9	미터링 단위	R	Single	필수	Integer	-	미터링 값 단위, **표** 53 참조

미터링 단위 리소스는 **표 53**과 같이 정의한다.

표 53 — 미터링 단위 리소스 값

값	의미
0	정의하지 않음
1	A (Current in Amperes (RMS))
2	Kelvin (Temperature)
3	Degrees Celsius (Relative temperature)
4	Voltage
5	J (Energy joule)
6	Hz (Frequency)
7	W (Real power in Watts)
8	m^3 (Cubic Meter)
9	VA (Apparent power)
10	VAR (Reactive power)
11	CosTheta (Displacement Power Factor)
12	V^2 (Volts squared)
13	A^2 (Amp squared)
14	VAh (Apparent energy)
15	Wh (Real energy in Watt-hours)
16	varh (Reactive energy)
17	Ah (Ampere-hours / Available Charge)
18	ft^3 (Cubic Feet)
19	ft^3/h (Cubic Feet per Hour)
20	m^3/h (Cubic Meter per Hour)
21	US gl (US Gallons)
22	US gl/h (US Gallons per Hour)
23	IMP gl (Imperial Gallons)
24	IMP gl/h (Imperial Gallons per Hour)
25	BTU
26	BTU/h
27	Liter
28	L/h (Liters per Hour)
29	PA(gauge)
30	PA(absolute)
31	Therm
32~	예약

7 보안 규격

7.1 개요

e-IoT 보안은 플랫폼과 게이트웨이, 게이트웨이와 디바이스와 플랫폼과 디바이스 구간에서의 안전한 채널 통신을 보장하기 위한 규격을 정의한다. e-IoT 보안은 OMA-LwM2M-Transport(Lightweight Machine to Machine Technical Specification: Transport Bindings Approved Version:1.1.1)의 5장 내용 중 인증서기반 DTLS 프로토콜을 적용하며, 대한민국 전자정부법 제 56조 및 시행령 제69조에 의거한 국가 및 공공기관의 중요한 정보 보호 사업을 수행하기 위해 필요한 내용을 정의한다.

7.2 보안 스팩 프로파일

본 절에서는 e-IoT의 인증서기반 DTLS보안에서 핸드쉐이크를 통한 협상과정을 지원한다. 본 표준에서는

아래와 같은 제한된 암호화 스위트(Cipher Suite)을 지원한다.

표 54에서 제시하는 암호화 스위트는 e-IoT의 개체별로 가용 암호알고리즘을 이용하여, 기밀성/무결성/인증의 주요 보안기능을 제공하기 위해 사용되는 스펙을 의미한다. 각각의 자세한 동작 방법은 참조 표준을 따른다.

표 54 — 암호화 스위트 (Cipher Suite)

IANA name	핵사코드	참조 표준
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	0xC0, 0xAE	RFC 7251
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	0xC0, 0x23	RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xC0, 0x2B	RFC 5289
TLS_ECDHE_ECDSA_WITH_ARIA_128_CBC_SHA256	0xC0, 0x48	RFC 6209
TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256	0xC0, 0x5C	RFC 6209

표 55 — 암호 표준 및 알고리즘 리스트

지원 암호 표준 및 알고리즘	구분	참조 표준
Datagram Transport Layer Security (DTLS)	프로토콜	RFC 6347
ECDHE (Elliptic Curve Diffe-Hellman Ephemeral)	키 교환	RFC 8442
ECDSA (Elliptic Curve Digital Signature Algorithm)	인증	—
AES 128 CCM 8 (AES with 128bit key in Counter with CBC-MAC node with an 8-Octet)	암호	RFC 6655, RFC 7251
AES 128 CBC (AES with 128bit key in Cipher Block Chaining mode)	암호	RFC 8442
ARIA 128 CBC (ARIA with 128bit key in Cipher Block Chaining mode)	암호	RFC 6209
AES 128 GCM (AES with 128bit key in Galois/Counter mode)	암호	RFC 5288
ARIA 128 GCM (ARIA with 128bit key in Galois/Counter mode)	암호	RFC 6209
SHA256 (Secure Hash)	해시	—

7.3 e-IoT 보안 초기설정

인증서기반 DTLS를 위한 초기 설정과정은 다양한 방법들이 가능하다. 예를 들면 OMA-LwM2M-Transport의 **5.2.4**에 설명한 Bootstrapping를 적용할 수 있다. KISA-GD-2014-0013에 따라서 다음과 같이 요구사항이 만족되어야 하며 본 규격에서는 초기설정의 구체적인 표준은 규격범위에 포함하지 않는다.

- 인증서 분배 주체는 신뢰할 수 있도록 인가되어야 한다.
- 안전한 저장 공간에 인증서가 저장되어야 한다.
- 인가된 수신자가 인증서를 수신해야 한다.

7.4 e-IoT와 DTLS 역할

플랫폼과 게이트웨이사이와 플랫폼과 디바이스 사이에서는 플랫폼이 DTLS 서버, 게이트웨이와 디바이스가 DTLS 클라이언트가 된다. 게이트웨이와 디바이스 사이에서는 게이트웨이가 DTLS 서버, 디바이스가 DTLS 클라이언트가 된다.

7.5 인증서 기반 보안 연관 성립 구조

인증서 기반의 보안 구조는 e-IoT 구성요소의 식별 및 인증 정보로 X.509 v3 인증서를 사용하며 DTLS 핸드쉐이크를 통해 교환한다. 그리고 전자 서명 검증을 통해 상호 인증을 수행한다.

a) 자격증명 설정(Credential Configuration)
   — 인증서, 인증서 체인, 공개키/비밀키 쌍 등의 자격 증명에 필요한 정보들을 설정한다.

b) 연관 설정(Association Configuration)
   — 보안 연관 핸드쉐이크 과정에 인증을 위해 필요한 식별자, 인증서, Trust Anchor 등의 보안 속성 정보를 설정한다.

c) 보안 연관 핸드쉐이크(Association Security Handshake)
   — 연관 설정 단계에서 설정한 보안 속성 정보를 이용하여 Association Handshake를 수행한다.
   — 인증서를 교환하여 인증서의 Path, Status를 검증한다. 선택적으로 인증서 체인을 전달할 수 있다. 전자서명을 이용하여 상호 인증을 수행한다. 타원곡선암호를 이용한 ECDSA를 사용할 수 있다.

e-IoT 규격에서 인증서 기반 타원곡선암호를 사용하기 위해서는 IETF RFC 5280의 X.509 v3 인증서 표준을 따르는 것을 권고한다.

인증서 모드의 구체적인 요구사항은 OMA_LwM2M_Transport의 **5.2.8.3**을 지원하며 인증서 기반 구조의 보안 연관 핸드쉐이크에서는 **표 54** 암호화 스위트를 따른다.

7.6 인증서 만료

인증서 사용을 위해서는 게이트웨이와 디바이스는 실시간 클럭 장치가 장착되어야 하고 기준 시간이 제공되어야 인증서 만료를 할 수 있다. 인증서 만료를 검출하기 위한 기준 시간에 대한 내용은 OMA_LwM2M_Transport의 **5.2.8.5**을 참조한다.

7.7 인증서 폐기

DTLS 클라이언트 역할을 수행하는 디바이스와 게이트웨이는 DTLS 서버역할을 수행하는 노드의 인증서의 폐기 시점을 결정할 수 있어야 한다. 구체적인 내용은 OMA_LwM2M_Transport의 **5.2.8.6**를 참조한다. 폐기된 인증서를 새로운 인증서로 교체하는 방법은 본 규격에 포함하지 않는다.

7.8 DTLS 에러 처리

네트워크상의 에러나 패킷 손실 등과 같은 다양한 이유로 DTLS 세션연결이 해제 될 수 있다. 이 경우 DTLS 에러의 종류를 인식할 수 있어야 하며, 이 에러를 처리하는 방법은 OMA-LwM2M-Transport의 **5.2.9**를 따른다.

부속서 A
(활용사례)

본 부속서는 에너지·전력 사물인터넷 표준의 산업적 활용 범례를 설명하는 것으로, 표준 내용의 일부이다.

A.1 e-IoT 산업적 활용 범례

사물인터넷은 ICT(정보통신기술)을 기반으로 사물과 사람, 사물과 사물간을 연결하여 정보를 전달하고 상호 통신하는 지능형 인프라 및 서비스 기술이다. 최근 많은 사물들이 인터넷과 연결되어 데이터를 생성하고 제어되면서 새로운 사용가치를 만들어내는 ICT 환경구축에 대한 요구가 증대되고 있다

특히 최근 산업 환경의 대규모화에 따른 에너지 가격과 환경오염 문제가 부각됨에 따라 친환경 저탄소 에너지 확산을 위한 그린 뉴딜 정책과 DNA(데이터, 네트워크, 인공지능) 혁신 생태계 구축을 위한 디지털 뉴딜 정책이 추진되고 있다. 정책 이행을 뒷받침하는 기술 중 하나로 사물인터넷 기술이 논의되고 있으며, 기존 에너지·전력 시스템과의 결합모델 발굴 및 에너지-IoT 융합서비스 개발 사업이 활발히 진행 중에 있다.

이 표준은 '디바이스-게이트웨이-네트워크-플랫폼-서비스'을 하나로 연결하는 표준 프레임워크를 제정하고 이를 토대로 에너지사업에 활용 가능한 모델을 개발할 수 있다. 현장에 활용 중이거나 적용 예정인 서비스는 다음과 같다.

a) 발전: 발전소 설비 안전 감시(비계 상태, 작업자 위치 등), 신재생 발전 및 상태 감시
b) 송전: 송전선로 및 철탑 감시(선로 상태, 항공장애등, 철탑변위, 누설애자)
c) 배전: 배전 시설물 감시 및 진단(변대주 기울기, 변압기 및 개폐기 상태), 굴착감시
d) 수용가: APT 수전설비 감시(전기품질(전류, 전압), 정전), 원격검침, 에너지관리
e) 기타: 통신망 단말관리, 가로등 원격감시

제정된 표준은 에너지·전력분야의 사물인터넷 환경구축에 있어 공통기술 활용하여 중복투자를 회피하고 다양한 제조사 참여를 유도함으로써 사용자들에게는 편의 서비스를 제공하고, 제조사, 서비스 사업자들에게는 다양한 영역의 생태계 구축과 신 사업 발굴의 기회를 제공할 수 있다.

A.2 e-IoT 적용 서비스 사례

e-IoT 시스템은 여러 사물인터넷 기술과 연계되어 전력 생산부터 사용단계까지 전력설비에 사용되는 기기들을 연결한다. 또한 각종 정보를 수집·분석하여 에너지 분야에 특화된 서비스를 제공할 수 있다. 즉, e-IoT 기술을 활용하면 에너지·전력분야의 非 ICT기술로 관리·운영 중인 시설을 e-IoT 시스템 기반으로 디지털화(Digitalization)할 수 있다.

그림 A.1은 전력에너지 생산부터 소비까지 단계별 전력에너지 시스템에서 활용 가능한 e-IoT 서비스 프로파일을 정의한다. 이러한 전력설비에 e-IoT를 적용함으로써 이상징후 사전감지 및 조치, 설비 모니터링 및 진단을 통해 효율적으로 시설물을 관리할 수 있다. 또한, 전력시스템은 국가시설물로서 신뢰성, 안정성 및 확장성을 고려해서 체계적으로 설계해야 한다. 즉, 디바이스와 시스템간 정보 교환을 위해 국제표준기반의 e-IoT를 적용하여 기술의 신뢰성 뿐만 아니라 상호운용성을 보장해야 한다.

본 표준을 기반으로 에너지·전력분야의 설비에 가장 많이 활용되는 서비스를 중심으로 e-IoT 서비스 절차 및 데이터 모델을 정의하고, 디바이스에서부터 표준화된 형태의 데이터를 플랫폼에 전달해 상호운용성을 확보한다면 향후 인공지능(AI)·데이터기반 서비스 확산에 기여 할 것으로 기대된다.

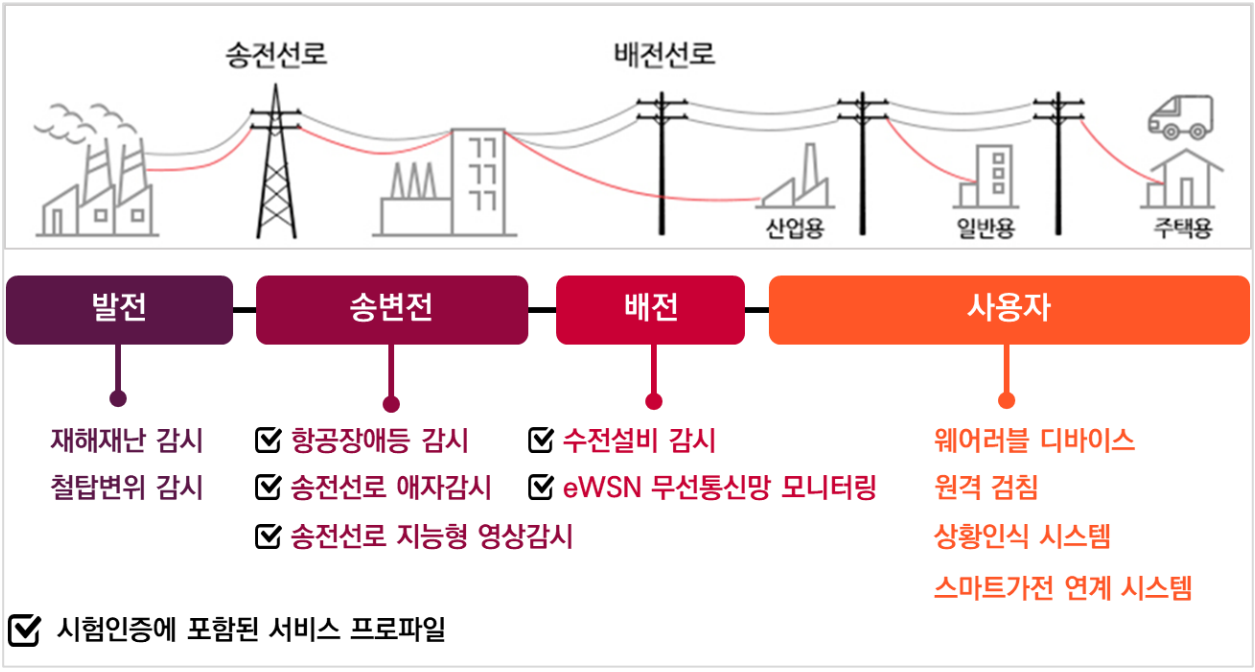

그림 A.1 — 에너지·전력분야 e-IoT 서비스 프로파일

A.2.1 항공장애등 원격감시 서비스

그림 A.2은 e-IoT을 적용한 항공장애등 원격감시서비스의 시스템 구성도를 나타낸다. 본 서비스는 송전철탑을 항공기로부터 보호하기 위한 목적으로 철탑에 설치된 항공장애등의 점멸상태, 배터리, 태양광 전압, 제어함 온도 등의 감시 기능을 가진다.

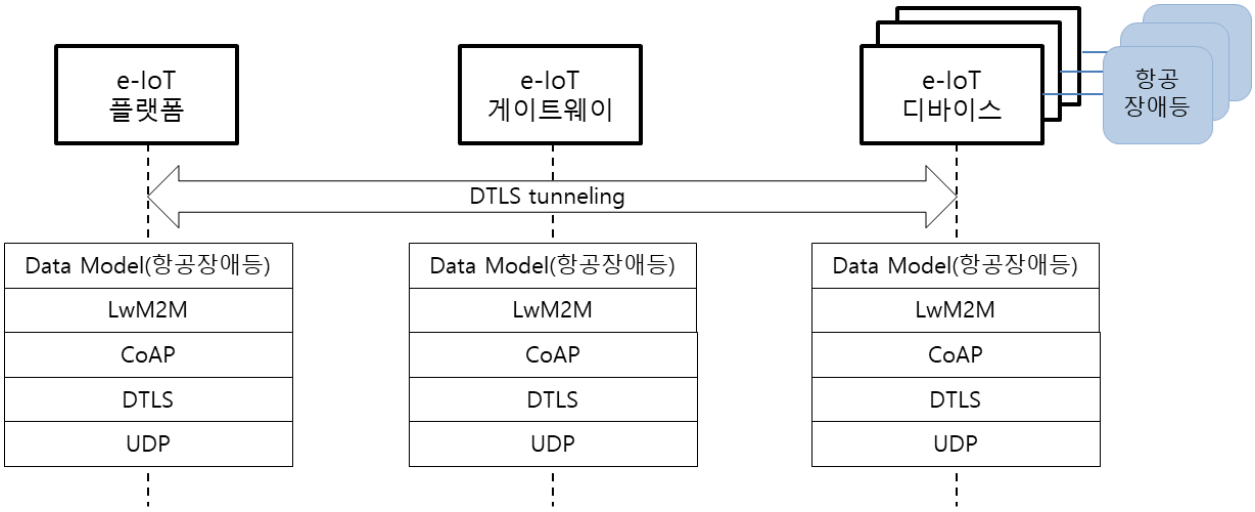

그림 A.2 — e-IoT 활용 서비스(항공장애등 원격감시)

A.2.2 항공장애등 원격감시 서비스 오브젝트-리소스 데이터 모델 예시

A.2.2.1 장애등 상태

표 A.1 — 장애등 상태 오브젝트

Name	Object ID	Object URN	Instances	Description
On/Off state	20022	urn:oma:lwm2m:x:20022	Multiple	항공장애등 관련 상태를 전달: Object Instance 0 ~ 7

표 A.2 — 장애등 상태 리소스

Resource Name	Resource ID	Access Type	Resource Instances	Resource Mandatory	Resource Type	Resource Range or Enumeration	Resource Units	Resource Description
Digital Input State	5500	R	Single	Mandatory	Boolean	-	-	항공장애등의 현재 상태 (0 or 1)
Digital Input Counter	5501	R	Single	Optional	Integer	-	-	디지털 Input의 변경 수 (0에서 1로)
Flicker	30013	R	Single	Optional	Integer	17..90	-	점멸되는 횟수(분당) 저광도: 60~90 중광도: 20~60 고광도: 17~20
On Time	5852	R, W	Single	Optional	Integer	-	s	On command가 보내진 후 시간
Cumulative On Time	30014	R	Single	Optional	Integer	-	s	On Time의 합
Threshold Time	30015	R, W	Single	Optional	Integer	8000	hr	램프수명 정의, 보조등 사용 시 참조값
Off Time	5854	R, W	Single	Optional	Integer	-	s	Off command가 보내진 후 시간
Application Type	5750	R, W	Single	Optional	String	-	-	센서의 응용타입 (ex. 항공장애등)
Target Device type	30000	R, W	Single	Optional	Integer	0..4	-	연계 설비 종류 (0: 없음, 1: 변압기, 2: 개폐기, 3: 철탑, 4:기타)
Target Device ID	30001	R, W	Single	Optional	String	-	-	연계 설비 ID
Target ID Format	30002	R, W	Single	Optional	Integer	0..1	-	연계 설비 ID 포맷 (0: OID, 1: Kepco ID)
Measurement Period	30003	R, W	Single	Optional	Integer	-	ms	센서의 계측 주기

참고문헌

다음 문서들은 이 표준의 이해를 돕기 위한 문서로서 특정 문서(발행일 및 판 번호 또는 개정 번호를 명시한 것)와 일반 문서로 구별된다.

— 특정 문서인 경우, 해당 판본 이후의 개정판은 적용되지 않는다.
— 일반 문서인 경우, 최신 판본이 적용된다.

[1] OMA, “OMA Lightweight (LwM2M) Object and Resources Registry”

[2] TTAE.IF-RFC7252 (2015), “제한된 환경에서의 응용 프로토콜(CoAP)”

[3] TTA TTAK.KO-101121: 에너지전력분야 사물인터넷(e-IoT) 제1부~제4부

[4] TTA TTAK.KO-101200: 에너지전력분야 사물인터넷(e-IoT) 상호운용성 시험규격

[5] TTA TTAK.KO-101205: DLMS와 LwM2M 프로토콜 간 데이터 연계 모델

KS X 3280:2021
해 설

이 해설은 이 표준과 관련된 사항을 설명하는 것으로, 표준의 일부는 아니다.

1 제정의 취지

이 표준은 에너지전력분야 사물인터넷을 활성화하기 위해 현장의 열악한 상황을 고려하여 손쉽게 운용될 수 있는 플랫폼과 게이트웨이, 게이트웨이와 디바이스, 플랫폼과 디바이스간 인터페이스에 대한 것으로 현장의 디바이스 상태 수집·감시·제어 서비스 제공을 목적으로 디바이스(센서 및 액추에이터)에서 측정된 정보를 원격에서 수집하고 제어할 수 있으며, 수집된 측정 정보를 플랫폼에 전달하는 기능을 제공하는 서비스 표준을 제공하는 것이다.

전력시스템은 국가기반시설로서 신뢰성, 안전성, 기업의 공적 책임 등이 요구되는 산업이다. 특히 제품 수명주기가 상대적으로 길며, 사고 발생시 국민안전과 산업피해가 매우 크기 때문에 중요시설로서 보호받고 관리, 운영되고 있다. 최근 신재생에너지 개발, 슈퍼그리드, 노후 전력설비 교체 등으로 전력망에 대한 수요는 날로 증가하고 있으며, 더 나아가 대형화, 밀집화 다기능화의 형태로 진화하고 있어, 이에 대한 관리 중요성이 갈수록 높아지고 있다.

전기설비의 고장징후를 미연에 감지, 사고를 방지하고 실시간으로 설비상태를 모니터링, 진단하기 위한 방안으로 사물인터넷 기술이 각광받고 있다. 이에 반해 전력시스템은 발전, 송변전, 배전, 수용가 등 전국 산지에 다양한 종류의 전력설비가 산재해 있기 때문에 소규모의 단위 시스템으로 구축, 운영됨으로써 IoT 수요증가 대비 관련 시장의 성장은 매우 더딘 상황이다.

특히 IoT 기술은 공통기술을 토대로 서비스별 특화함으로써 제품 고도화가 이루어지는 특성을 갖지만, 에너지 전력분야에서는 제조사별 단위 시스템으로 구축됨으로서 기기 및 시스템간 정보연계가 어렵고, 제품간 상호운영성 확보가 되지 않음에 따라 구축비용 증가 및 유지보수, 서비스 확대 등의 문제점으로 지적되고 있는 상황이다.

이에 이 표준에서는 상기 문제를 해결하고자, 디바이스 – 게이트웨이 – 플랫폼 전주기 관점에서의 구성요소 간 서비스 절차 및 리소스 정보모델을 정의하고 표준화된 형태의 데이터를 전달함으로써 다양한 구성요소 간 상호운용성을 보장하고, 이를 토대로 연결 및 정보교환, 활용할 수 있는 방법을 제공하고자 한다.

2 제정의 경위

이 표준은 한국정보통신기술협회의 단체표준을 기반으로 표준 초안이 만들어졌으며, 방송통신표준심의회 사물인터넷 전문위원회에서 원안을 수정 보완하고 상세한 검토를 통해 초안을 완성하고 방송통신표준으로 제안하였다.

3 특허권 등에 관한 사항

이 표준과 관련하여 지식재산권 확약서 정보는 다음과 같다.

a) 지식재산권 확약서
   — 발명의 명칭: 에너지 계량 데이터 관리 장치, 이의 방법, 이 방법을 저장한 컴퓨터 판독 가능

저장 매체
— 권리자의 성명: 한국전력공사
— 등록번호: 10-2062636-0000
— 등록(출원) 연월일: 2019년 12월 30일
— 실시조건: 지식재산권을 합리적 조건하에 비차별적으로 실시
— 특허문의: 한국전력공사 대표전화 061-345-3114

b) 지식재산권 확약서
— 발명의 명칭: 검침 데이터 수집 시스템, 방법, 이를 저장한 기록 매체
— 권리자의 성명: 한국전력공사
— 등록번호: 10-1999314-0000
— 등록(출원) 연월일: 2019년 7월 5일
— 실시조건: 지식재산권을 합리적 조건하에 비차별적으로 실시
— 특허문의: 한국전력공사 대표전화 061-345-3114

c) 지식재산권 확약서
— 발명의 명칭: 사물인터넷 단말 운영방법
— 권리자의 성명: 한국전력공사
— 출원번호: 10-2018-0113262
— 등록(출원) 연월일: 2018년 9월 20일
— 실시조건: 지식재산권을 합리적 조건하에 비차별적으로 실시
— 특허문의: 한국전력공사 대표전화 061-345-3114

KS X 3280:2021

KSKSKS
KSKSK
KSKS
KSK
KS
KSK
KSKS
KSKSK
KSKSKS

Interface of internet of things in energy-electric power domain(e-IoT)

ICS 19.020