

CraneWireRope

GW 통신 프로토콜

AIOTION



한글판(Korean)

CraneWireRope GW 통신 프로토콜

본 문서는 *AIOTION*® 제품의 게이트웨이 통신에 대해 정의한 문서입니다.

문서 정보

구분	소속	성명	날짜	비고
Owner	AIOTION팀	박상재		
작성자	AIOTION팀	박상재	2019-02-18	
동료검토자	AIOTION팀	권태엽		
	AIOTION팀	유성수		
승인자	AIOTION팀	남상훈		
버전	발행일	상태		보존연한
V1.0	2019-02-18	Public/Draft		5년

개정 이력

개정일자	버전	개정내역	작성자	확인자
2019-02-18	V1.0	최초 작성	박상재	남상훈
2019-03-19	V1.1	메시지 포맷 수정	박상재	유성수
2019-03-22	V1.2	LED 제어 명령 수정	박상재	유성수

목 차

1.	개 요	4
1.1.	목적	4
1.2.	용어 및 약어	4
1.3.	참조문서	4
2.	AIOTION-CraneWireRope Gateway 개요	5
2.1.	CraneWireRope 통신 구성	5
2.2.	Message Format	6
2.2.1.	등록 요청 (Topic : "cwr/mfl/{fwld}/contract/req")	6
2.2.2.	제어 명령 요청 (Topic : "cwr/mfl/{fwld}/command/req")	7
2.2.3.	제어 명령 응답 (Topic : "cwr/mfl/{fwld}/command/res")	9
2.2.4.	데이터 전송 (Topic : "cwr/mfl/{fwld}/data")	9
2.2.5.	센서노드 상태 전송 (Topic : "cwr/mfl/{fwld}/health")	10
2.2.6.	엔코더 데이터 전송 (Topic : "cwr/mfl/{fwld}/encoder")	11

1.개 요

1.1. 목적

AIOTION-CraneWireRope 제품의 게이트웨이와 센서노드 연동 펌웨어와의 통신 구성과 메시지 포맷을 설명하는 문서이다.

게이트웨이간의 구성도를 설명하고 상호 통신 방식 및 메시지 포맷이 정의되어 있다

1.2. 용어 및 약어

본 문서에서 사용하는 용어와 약어는 다음과 같다.

용어 및 약어	의미
CWR	AIOTION-CraneWireRope
GW	Gateway
FW	Firmware

1.3. 참조문서

본 문서에서 참조하고 연관있는 문서목록은 다음과 같다.

문서명	설명

2. AIOTION-CraneWireRope Gateway 개요

2.1. CraneWireRope 통신 구성

CWR GW의 기본 구성은 다음과 같다. 센서노드-펌웨어 구간은 900Hz 대역 무선통신, 엔코더-펌웨어구간은 시리얼 통신, 펌웨어-GW구간은 MQTT 통신, GW-AP 구간은 LTE가 사용된다.

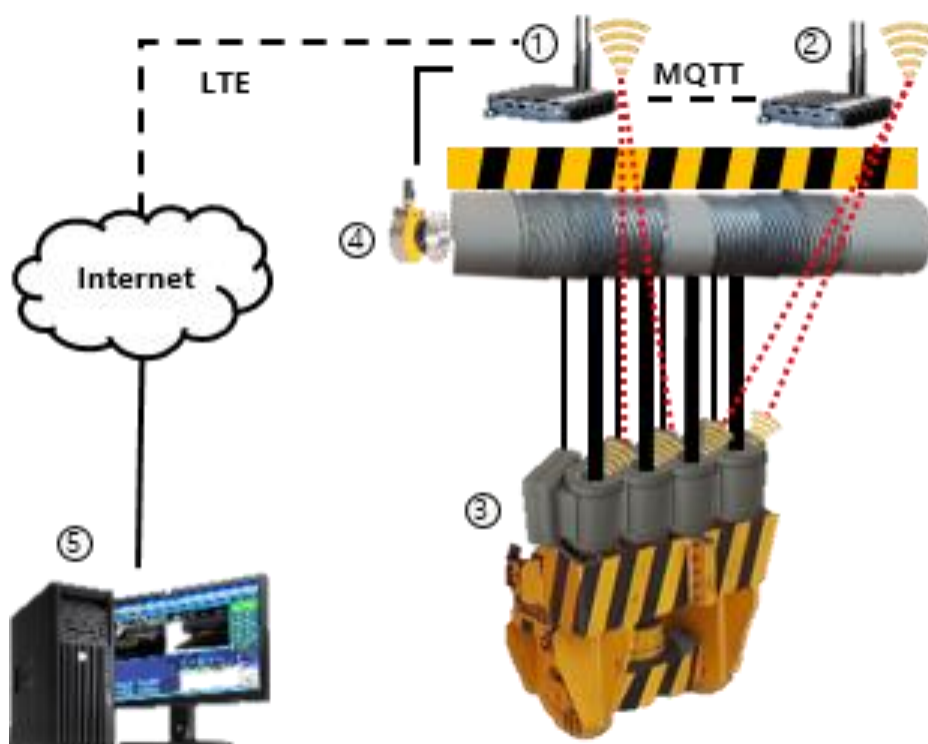


그림 1 AIOTION-CWR 구성

- ① CWR GW Software + 센서노드 연동 펌웨어
- ② 센서노드 연동 펌웨어
- ③ 센서노드
- ④ 엔코더
- ⑤ CWR Centric AP

센서연동 펌웨어가 1:N 매칭된 센서노드와 엔코더의 데이터를 수집하고 GW 와 MQTT 통신을 통해 이를 전달한다. 1개의 GW는 최대 2개의 센서연동 펌웨어와 통신 가능하고 1개의 센서연동 펌웨어는 최대 2개의 센서노드와 통신 가능하다. CWR GW는 LTE를 통해 AP에 데이터를 전달한다.

2.2.Message Format

CWR GW는 센서연동 펌웨어와의 통신을 위해 MQTT 프로토콜을 지원한다. 메시지 형식은 JSON 형식을 사용한다. 펌웨어는 MQTT Topic에 고유ID를 할당하여 사용한다. 메시지 포맷은 Topic별로 구성된다. TimeStamp Field는 UNIX TimeStamp (10자리) 형식을 준수한다.

Topic	Description
cwr/mfl/{fwld}/contract/req	Firmware 가 GW 에게 연결 요청
cwr/mfl/{fwld}/command/req	GW 가 Firmware 에게 명령 요청
cwr/mfl/{fwld}/command/res	Firmware 가 GW 에게 명령 응답
cwr/mfl/{fwld}/data/push	Firmware 가 GW 에게 데이터 전송
cwr/mfl/{fwld}/health/push	Firmware 가 GW 에게 생존정보 전송
cwr/mfl/{fwld}/encorder/push	Firmware 가 GW 에게 엔코더 데이터 전송

표 1 CWR MQTT Topics

2.2.1. 등록 요청 (Topic : "cwr/mfl/{fwld}/contract/req")

센서연동 펌웨어는 (구동시 / 네트워크 연결시 / 재시작시) GW에 contract 요청 메시지를 전송한다.

Format

GW contract 요청 메시지 포맷
<pre>{ "msgId" : "195841595230", "timeStamp" : "1551092460" "sensorNode" : [{ "snId" : "bbb111", "chCount" : 2 }, { "snId" : "bbb222", "chCount" : 4 } }]</pre>

}
sensorNode : Firmware가 관리하는 센서노드 목록
snId : 센서노드 고유 ID
chCount : 센서노드 채널 개수

2.2.2. 제어 명령 요청 (Topic : "cwr/mfl/{fwld}/command/req")

CWR-GW는 센서연동 펌웨어에 제어 명령을 요청할 수 있다. 요청 타겟은 Type으로 구분한다.

Type	구분	value	Description
led	RGB-LED 제어	ron	LED 의 상태를 rOn 으로 변경 (빨강)
		gon	LED 의 상태를 gOn 으로 변경 (초록)
		bon	LED 의 상태를 bOn 으로 변경 (파랑)
		off	LED 의 상태를 Off 으로 변경
		rflash	LED 의 상태를 반짝임 으로 변경 (빨강)
		gflash	LED 의 상태를 반짝임 으로 변경 (초록)
		bflash	LED 의 상태를 반짝임 으로 변경 (파랑)
status	동작 제어	active	센서노드의 상태를 Active 로 변경
		sleep	센서노드의 상태를 Sleep 으로 변경

표 4 제어 명령 설명

1) LED 제어

센서노드의 LED 제어 명령을 요청한다. 센서노드에 설치되는 LED의 개수는 변할 수 있으며 각 LED 는 value field의 index로 구분한다

Format

LED 제어 명령 메시지 포맷
LED 제어 예시) SensorNode#3의 LED#2 의 상태를 빨강으로 On 명령 <pre>{ "msgId" : "195841595230", "timeStamp" : "1551092460"</pre>


```

    "type" : "led",
    "nodeId" : ["bbb111"],
    "value" : "ron"
  }

```

2) 동작 제어

센서노드의 동작 제어 명령을 요청한다. 센서연동 펌웨어는 active 명령을 수신하면 소속된 센서노드들에게 명령을 전달하고 센서노드들은 센싱을 시작하고 데이터를 푸쉬한다. Sleep 명령을 수신하면 센서노드의 센싱을 중지하고 센서노드의 메모리에서 미전송한 데이터를 모두 전송한 후 센서노드를 sleep 상태로 전환한다.

동작 제어 명령 메시지 포맷
<p>동작 제어 예시)</p> <p>SensorNode#3, SensorNode#4로 active 명령</p> <pre> { "msgId" : "195841595230", "timeStamp" : "1551092460" "type" : "status", "nodeId" : ["bbb111", "bbb222"], "value" : "active" } </pre>

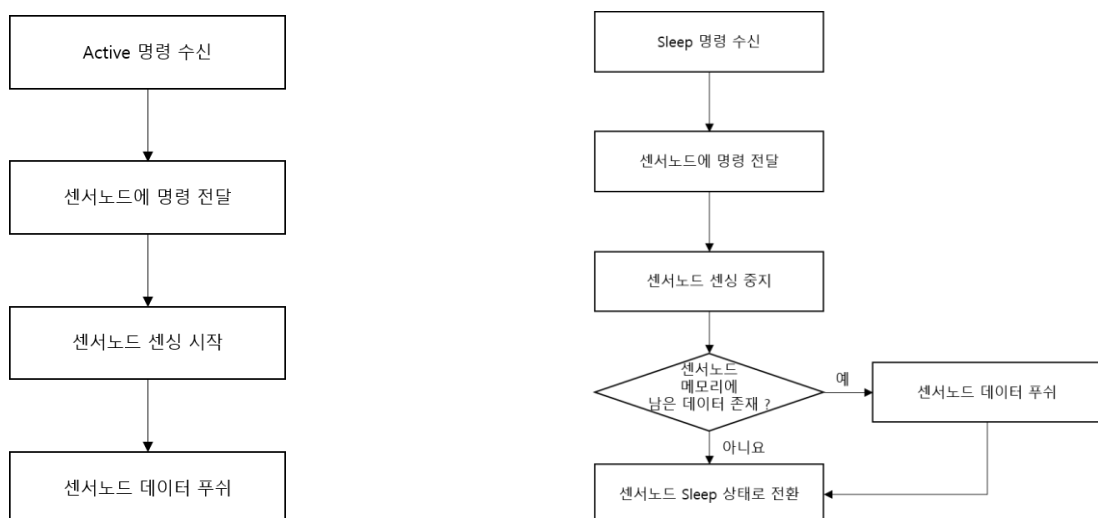


그림 2 active 수신시 순서도

그림 3 sleep 명령 수신시 순서도

2.2.3. 제어 명령 응답 (Topic : "cwr/mfl/{fwld}/command/res")

CWR GW의 제어 명령에 따른 결과 메시지를 센서연동 펌웨어에서 전송한다. 실패 시 메시지 field에 실패 원인을 같이 전송한다.

Format

제어 응답 메시지 포맷
<pre>{ "msgId" : "195841595230", "timeStamp" : "1551092460" "result" : "fail", "message" : "SN#3 timeout" }</pre>

구분	Description
success	제어 성공
fail	제어 실패

표 5 result Field 설명

2.2.4. 데이터 전송 (Topic : "cwr/mfl/{fwld}/data")

센서연동 펌웨어는 GW로부터 수신한 active명령 메시지의 timeStamp를 currentTime으로 세팅 하며 이후의 모든 채널의 데이터를 1sec 주기로 푸쉬 전송한다.

Format

데이터 푸쉬 메시지 포맷
<pre>{ "msgId" : "195841595230",</pre>

```

"timestamp" : "1551092460"
"chData" : [{
    "nodeId" : "bbb111"
    "rawMfl" : [[134.56, 122., ... ],
                [134.56, 122., ... ]]
},
{
    "nodeId" : "bbb222"
    "rawMfl" : [[134.56, 122., ... ],
                [134.56, 122., ... ],
                [134.56, 122., ... ],
                [134.56, 122., ... ]]
}]
}

```

2.2.5. 센서노드 상태 전송 (Topic : "cwr/mfl/{fwld}/health")

센서연동 펌웨어는 센서노드의 상태 데이터를 3600 sec 주기로 푸쉬 전송한다. 센서노드의 가용 상태를 Availabillity field에서 TRUE, FALSE 로 표시하며 battery field 는 각 센서노드의 전압값을 표시한다.

Format

센서노드 상태 전송 메시지 포맷
<pre> { "msgId" : "195841595230", "timestamp" : "1551092460" "health" : [{ "nodeId" : "bbb111", "availabillity" : TRUE, "battery" : 3.5 }, { "nodeId" : "bbb222", "availabillity" : FALSE, "battery" : 2.8 }] } </pre>

```

    }}
}

```

2.2.6. 엔코더 데이터 전송 (Topic : "cwr/mfl/{fwld}/encoder")

센서연동 펌웨어는 엔코더의 데이터를 1 sec 주기로 푸쉬 전송한다.

Format

데이터 푸쉬 메시지 포맷
<pre> { "msgId" : "195841595230", "timeStamp" : "1551092460" "enData" : [1234.56, 1235.23, ...] } </pre>

구분	푸쉬 주기
contract	전원 On 이후 (재시작시에도 적용)
chData	1 sec (active 명령을 받은 경우)
health	3600 sec
encoder	1 sec

표 7 Default 푸쉬 주기 설명