

□ 교통신호제어센터의 교차로 신호정보 제공 방안

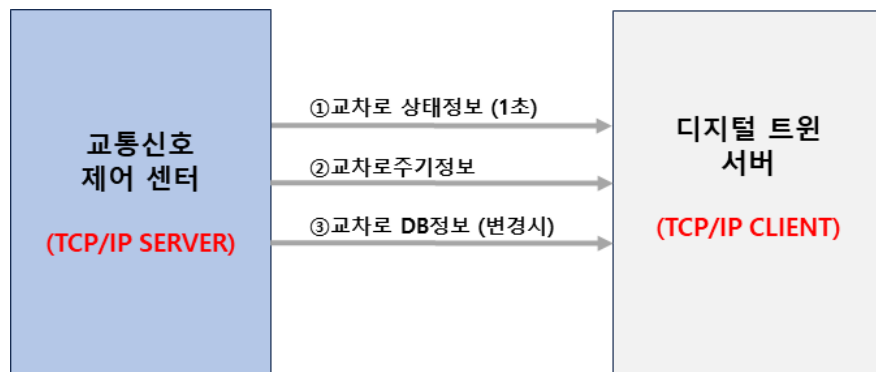
1. 개요

1) 본 문서는 성남시 신호제어 서버에서 외부시스템에 신호정보 제공을 위한 인터페이스를 정의한다.

2) 신호정보제공 대상 외부시스템은 다음과 같다

- 디지털 트윈 서버
- 긴급차량관제 서버
- 스마트교차로 서버
- ITS 센터

3) 연계 구성



2. 통신 방식

1) 통신 방식 : TCP 소켓 통신

[보안정책상 UDP 소켓으로 변경여지 있습니다....]

Server 측	Client 측	포트	규격
교통신호제어센터	외부시스템	7072 (TBD)	TCP/IP

2) Byte Ordering : Big-Endian

- 최상위 바이트(MSB)를 먼저 보내고, 최하위 바이트(LSB)는 맨나중에 보냄

3) 데이터 형식 : Byte 형식

4) 문자 인코딩 : UTF-8

5) 포트번호 : 7072

3. 프레임 구조

- 1) 본 시스템의 프레임은 "HEADER + DATA"로 구성하며, "HEADER"의 구성과 "COMMAND" 목록은 다음과 같다.

■ HEADER 구성 (10 BYTES)

항목	설명	데이터크기	데이터 유형
STX1	통신프레임의 시작 부호 1 (0x7E)	1	BYTE
STX2	통신프레임의 시작 부호 2 (0x7E)	1	BYTE
SEQUENCE	순차번호 (0 ~ 255)	1	BYTE
TIME	현재시각 (32BIT)	4	BYTE
COMMAND	명령코드	1	BYTE
DATA LENGTH	데이터 프레임의 BYTE 길이	2	BYTE

■ COMMAND 목록

번호	COMMAND	설명	송수신 방향	비고
1	0xF2	교차로 신호운영현황 전송	신호서버 → 외부서버	
2	0xF3	교차로 신호운영현황 응답 (ACK)	신호서버 ← 외부서버	
3	0xF4	교차로 주기정보 전송	신호서버 → 외부서버	
4	0xF5	교차로 주기정보 응답 (ACK)	신호서버 ← 외부서버	
5	0xF6	교차로 DB 전송	신호서버 → 외부서버	
6	0xF7	교차로 DB 응답 (ACK)	신호서버 ← 외부서버	
7	0xF9	신호정보제공 요청	신호서버 ← 외부서버	예비

- 신호정보제공 요청시 사전 등록된 서버에 한하여 신호정보제공 (예정)

■ 전체 메시지 구성 (HEADER + DATA)

HEADER						DATA
STX1	STX2	SEQ.	TIME	COMMAND	LENGTH	DATA
1 byte	1 byte	1 byte	4 byte	1 byte	2 byte	[Length] Bytes

4. 데이터 상세 구조

1) 교차로 신호운영현황 정보전송

- 1초 단위의 교차로 신호운영현황 정보
- 정보전송 **“교차로 시작번호”**부터 N개의 교차로를 연속적으로 전송한다

BYTE	항목	BIT	설명			
1	교차로 번호	16 BIT	정보전송 교차로 시작번호 (1 - 9999)			
2						
3	현시 코드	7 - 5	RING A의 PHASE (0 ~ 7)			
		4 - 0	RING A의 STEP (0 ~ 31)			
4		7 - 5	RING B의 PHASE (0 ~ 7)			
		4 - 0	RING B의 STEP (0 ~ 31)			
5	제어기 운영상태	7	제어기 통신상태	1 : FAIL, 0 : 정상		
		6	운용 맵번호	0 : 일반제, 1~5 : 시차제, 6 : 전용맵		
		5				
		4	등기종류	1 : 4색등, 0 : 3색등		
		3				
		2			교통신호기 운영모드	0 : SCU 고정주기 모드 1 : 감응하지 않는 OFFLINE 제어모드 2 : 감응되는 OFFLINE 제어모드 4 : 감응되는 온라인 제어모드 5 : 감응하지 않는 온라인 제어모드
		1				
0						
6	제어기 상태	7	RING 운영방식	1 : DUAL-RING, 0 : SINGLE-RING		
		6	현시유지	1 : ON, 0 : OFF		
		5	우선신호	1 : 서비스 중, 0 : OFF		
		4	전이	1 : 전이 중, 0 : OFF		
		3	감응	1 : 감응, 0 : 정상		
		2	소등	1 : 소등, 0 : 정상		
		1	점멸	1 : 점멸, 0 : 정상		
		0	수동	1 : 수동, 0 : 정상		
7		-	주기 COUNT	0 ~ 255 초		
8		-	현 주기	0 ~ 255 초		
9		-	연동	실제 계측 OFFSET		
10		-	A-ring 이동류번호	1 - 21		
11		-	B-ring 이동류번호	1 - 21		
12 - 20	교차로 시작번호+1 신호운영정보 (9 Bytes)					
21 - 29	교차로 시작번호+2 신호운영정보 (9 Bytes)					
	...					
...	교차로 시작번호+N 신호운영정보 (9 Bytes)					

- 이동류번호는 단위시험 과정에서, 삭제 가능하다면 삭제 검토
- day plan table에 이동류 번호 지정되어 있어, 현시번호로 확인 가능

이동류번호는 시차제를 반영한 신호현시가 가능하도록 하기 위함임. 만약 day plan table이 실시간으로 조회 가능하다면 삭제 가능하지만 그렇지 않으면 삭제 불가능.

2) 교차로 주기정보 : 주기 종료 후 Split 정보 제공

- 주기 종료 후 Ring-A, Ring-B의 운영한 현시정보를 전송한다.
- 정보전송은 N개의 교차로 정보를 연속하여 전송이 가능하다

BYTE	항목	BIT	설명
1	교차로 번호	16 BIT	정보전송 교차로번호 (1 - 9999)
2			
3 - 10	RING-A 운영시간	8 Bytes	PHASE 1 - 8
11 - 18	RING-B 운영시간	8 Bytes	PHASE 1 - 8
	2번째 교차로 번호/ Ring-A / RING-B 운영시간 (18 Bytes)		
	3번째 교차로 번호/ Ring-A / RING-B 운영시간 (18 Bytes)		
	4번째 교차로 번호/ Ring-A / RING-B 운영시간 (18 Bytes)		
			

3) 교차로 DB 정보 : 교차로 DB 변경시 제공 (JSON 표기)

① 교차로 WEEK PLAN

■ JSON 표기

WEEK PLAN JSON 표기	WEEK PLAN data 설명
<pre>{ "lcid" : number, "type" : "weekplan", "data" : number[], }</pre>	data : 7 plan_no plan_no = 1 ~ 10

■ week plan table 정의

일요일	월요일	화요일	수요일	목요일	금요일	토요일
0	0	0	0	0	0	0

② 교차로 DAY PLAN

■ JSON 표기

DAYPLAN JSON 표기	DAYPLAN data 설명
<pre>{ "lcid" : number, "type" : "dayplan", "plan" : { "plan_no": number, "redYel": number[], "data": number[], } }</pre>	plan_no = 1 ~ 10 redYel : [A_flow_no, A_red_time, A_yellow_time, B_flow_no, B_red_time, B_yellow_time,] data : [hour, min, cycle, offset, split[16]]

- 이동류번호를 이곳에 넣어 대체하겠다는 의견인데, 그렇다면 이 변경사항을 업데이트되는 최소 주기가 얼마인지? 만약 하루 단위로 업데이트를 하는거라면, 의미가 없다. (시단위, 분단위도 의미가 없고 초단위여야 의미가 있다.) 만약 이동류번호가 변경되었을 때, 그 순간 (1초 내로) 변경된 값이 테이블에 반영된다면, 변경되었다는 신호가 외부서버에 즉각 전달된다는 가정 하에 신호생성이 가능하다. 그런데 이정도로 많은 데이터라면 1초 내로 업데이트하기는 힘들 것도 같다. 혹시, 변경사항이 있을 때에만 업데이트를 하는 거라서 1초 내로 업데이트가 가능한 것인지?

- 황색시간과 적색시간이 반영된 것은 좋은데, 가장 기본적인 녹색시간이 없다. 녹색시간 혹은 현시시간 (=녹+황+적)이 있어야 한다.

■ FLOW/RED/YELLOW

현시	A-ring			B-ring		
현시 1	이동류번호	red_time	yellow_time	이동류번호	red_time	yellow_time
현시 2						
현시 3						
현시 4						
현시 5						
현시 6						
현시 7						
현시 8						

■ DAY PLAN TIME Table

[illegible]

③ 교차로 HOLIDAY PLAN

■ JSON 표기

HOLIDAY PLAN JSON 표기	HOLIDAY PLAN data 설명
<pre>{ "lcid": number, "type": "holidayplan", "data": number[], }</pre>	[month, day, plan_no] * 30 plan_no = 1 ~ 10

■ HOLIDAY Plan Table

	월	일	계획번호
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0

④ SIGNAL MAP (Ring-A , Ring-B)

■ JSON 표기

SIGNAL MAP JSON 표기	SIGNAL MAP data 설명
<pre>{ "lcid": number, "type": "signal_map", "data": { "map_no" : number, "a_ring" : number[], "b_ring" : number[], }, }</pre>	map_no = 1 - 6 [output[16], min, max, eop] * 32 step

■ SIGNAL MAP Table

		LSU1		LSU2		LSU3		LSU4		LSU5		LSU6		LSU7		LSU8		LSU9		LSU10		LSU11		LSU12		LSU13		LSU14		LSU15		LSU16				
		G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	MIN	MAX	EOP		
A	1																															0	0	☐		
	2																															0	0	☐		
	3																															0	0	☐		
	4																															0	0	☐		
	5																															0	0	☐		
	6																															0	0	☐		
	7																															0	0	☐		
	8																															0	0	☐		
	9																															0	0	☐		
	10																															0	0	☐		
	11																															0	0	☐		
B		LSU1		LSU2		LSU3		LSU4		LSU5		LSU6		LSU7		LSU8		LSU9		LSU10		LSU11		LSU12		LSU13		LSU14		LSU15		LSU16				
		G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	G2	G1	MIN	MAX	EOP		
	1																															0	0	☐		
	2																															0	0	☐		
	3																															0	0	☐		
	4																															0	0	☐		
	5																															0	0	☐		
	6																															0	0	☐		
	7																															0	0	☐		
	8																															0	0	☐		
	9																															0	0	☐		
10																															0	0	☐			
11																															0	0	☐			

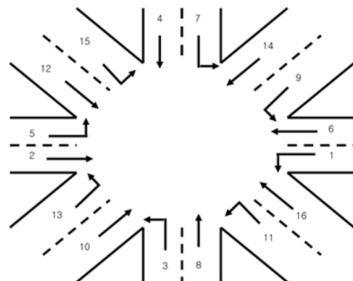
⑤ 교차로 기반정보 구성

■ JSON 표기

교차로 기반정보 구성 JSON 표기	교차로 기반정보 구성 data 설명
<pre> { "lcid": number, "type": "geo_map", "intName": string, // 교차로명 "intType": number, // 교차로유형 (0: 주교차로, 1: 연등교차로) "lcType": number, // 제어기 유형 (1: 2004년형, 2: 2010 년형, 3: 2023년형) "lampType": number, // 등기 유형 (1: 4색등, 2: 3색등) "mainIntNo": number, // 연등교차로인 경우 주교차로 번호 "groupNo": number, // 그룹번호 "nodeNo": number, // 교차로 노드 ID "intLat": number, // 교차로 위도 (ex : 36.123500) "intLng": number, // 교차로 경도 (ex : 127.123500) "ppcType": number, // PPC 유형 (0 : 설치안됨, 1 : 현장방식, 2 : 중앙제어방식) "mainP": number, // 주현시번호 (1 - 8) "flow": { "num": number, // 이동류 번호 "geoXY": number[], // 화살표 시작점 위도/경도, 중간점 위도/경도, 종료점 위도/경도, } }</pre>	

■ 이동류 번호

이동류 번호	이동류 방향	설명	이동류 번호	이동류 방향	설명	이동류 번호	이동류 방향	설명
1		좌회전 (동 → 남)	9		좌회전 (북동 → 남동)	17	·	보행신호
2		직진 (서 → 동)	10		직진 (남서 → 북동)	18	·	신호 없음
3		좌회전 (남 → 서)	11		좌회전 (남동 → 남서)	21	·	신호우회전
4		직진 (북 → 남)	12		직진 (북서 → 남동)			
5		좌회전 (서 → 북)	13		좌회전 (남서 → 북서)			
6		직진 (동 → 서)	14		직진 (북동 → 남서)			
7		좌회전 (북 → 동)	15		좌회전 (북서 → 북동)			
8		직진 (남 → 북)	16		직진 (남동 → 북서)			



A				
	6	5	16	15
B				
	2	1	12	11
A				
	8	7	14	13
B				
	4	3	10	9