**현대 자동차 PoC 미들웨어**

**응용서비스 연계 프로토콜 정의서**

|  |  |
| --- | --- |
| **문서코드** | XX-XX-XX-XX |
| **버전** | V0.1 |
| **작성일** | 2019-06-04 |
| **작성자** | 강 대현 |

**개정 이력**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **버전** | **개정일자** | **¹개정사유** | **²개정내역** | **개정자** | **승인자** |
| **V0.1** | **2019/06/04** | **추가** | **최초 설계자료 작성** | **강대현** |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**¹개정 사유 : 개정 또는 개정 내용이 이전 문서에 대해 추가/수정/삭제인지 선택 기입**

**²개정 내역 : 개정이 발생하는 페이지 번호와 변경 내용을 기술**

목차

[1. PLC수집 시작 프로토콜 3](#_Toc11317960)

[1.1. PLC수집 시작 Request 프로토콜(응용서비스 -> M/W) 3](#_Toc11317961)

[1.2. PLC수집 시작 Response 프로토콜(M/W -> 응용서비스) 5](#_Toc11317962)

[2. PLC 수집 정지 프로토콜 7](#_Toc11317963)

[2.1. PLC수집 정지 Request 프로토콜(응용서비스 -> M/W) 7](#_Toc11317964)

[2.2. PLC수집 정지 Response 프로토콜(M/W -> 응용서비스) 7](#_Toc11317965)

[3. PLC 수집 조회 프로토콜 7](#_Toc11317966)

[3.1. PLC수집 조회 Request 프로토콜(응용서비스 -> M/W) 7](#_Toc11317967)

[3.2. PLC수집 조회 Response 프로토콜(M/W -> 응용서비스) 7](#_Toc11317968)

[4. PLC 수집 일괄정지 프로토콜 7](#_Toc11317969)

[4.1. PLC수집 일괄정지 Request 프로토콜(응용서비스 -> M/W) 8](#_Toc11317970)

[4.2. PLC수집 일괄정지 Response 프로토콜(M/W -> 응용서비스) 8](#_Toc11317971)

[5. PLC 수집 정보 Publish 프로토콜 8](#_Toc11317972)

[5.1. PLC수집 정보 Publish 프로토콜(M/W -> 응용서비스) 9](#_Toc11317973)

# PLC수집 시작 프로토콜

* 응용서비스와 미들웨어 간 PLC수집 시작 명령을 위한 MQ(REQ-RES)기반의 프로토콜로 해당 명령 수신 시, 공정 정보 수집을 위한 미들웨어 Handler 시작과 정보 전달을 위한 MQ PUB 서버 기동 및 Publish를 개시(Event Context)한 후 처리 결과(Response)를 응답한다.
* 전송 데이터는 JSON표준 형식으로 전송
* 전송 문자 타입은 UTF-8 인코딩 사용.
* 전송 종료 코드는 CR LF ('\r\n')를 사용.

## PLC수집 시작 Request 프로토콜(응용서비스 -> M/W)

* Request JSON 스키마

{

“jsonrpc” : “2.0”,

“method” : “start”,

“id” : {“type” : “string”},

“param” : {

“protocol.version” : {“type” : “string”},

“event.id” : {“type” : “string”},

“plc.ip” : {“type” : “string”},

“plc.port” : {“type” : “string”},

“polling.period” : {“type” : “string”},

“publish.port” : {“type” : “string”},

“items” : [

{ “key” : {“type” : “string”},

“device.code” : {“type” : “string”},

“device.num” : {“type” : “string”},

“device.score” : {“type” : “string”}

}]

}

}\r\n

* Request 항목 상세

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **항목명** | | | **항목명(한글)** | **필수** | **설명** | **비고** |
| 1 | jsonrpc | | | Json rpc 버전 | O | Json rpc 프로토콜 버전 | 2.0 |
| 2 | method | | | 기능명 | O | 처리 | start |
| 3 | id | | | 통신ID | O | Req-Res 통신 ID |  |
| 4 | param | | | 파라미터 | O | 기능별 입력 파라미터 | Object |
| 5 |  | protocol.version | | 프로토콜 버전 | O | 프로토콜 버전 |  |
| 6 | event.id | | 이벤트 ID | O | 명령처리의 이벤트ID | 수집 시작/정지 및 조회 처리에 대한 Identifier로 사용 |
| 7 | plc.ip | | PLC 아이피 주소 | O | PLC연결을 위한 IP Address |  |
| 8 | plc.port | | PLC 아이피 포트 | O | PLC연결을 위한 Port Number |  |
| 9 | polling.period | | 수집주기 | O | 정보 수집 주기(s) |  |
| 10 | publish.port | | 수집정보 통신 포트 |  | 수집정보 전달을 위한 MQ 통신 포트 |  |
| 11 | items | | 아이템 정보 배열 | O | 정보 수집을 위한 아이템 정보 배열 | Array |
| 12 |  | key | 아이템명 | O | 수집 항목의 이름 |  |
| 13 |  | device.code | PLC 디바이스 코드 | O | PLC 디바이스 영역에 따른 코드 | \*액세스 가능 디바이스 일람 참조 |
| 14 | device.num | PLC 디바이스 번호 | O | PLC 디바이스의 메모리 번호(16진수 어드레스) | \*액세스 가능 디바이스 일람 참조 |
| 15 |  | device.score | PLC 디바이스 스코어 | O | PLC 디바이스의 메모리 스코어 | \*최대 사용가능 스코어 참조 |

* Request JSON 샘플

{

“jsonrpc” : “2.0”,

“method” : “start”,

“id” : “1”,

“param” : {

“protocol.version” :”1.0”,

“event.id” : “event1”,

“plc.ip” : “127.0.0.1”,

“plc.port” : “8192”,

“polling.period” : “3”,

“publish.port” : ”5000”,

“items” : [

{ “key” : “lot”,

“device.code” : “D\*”,

“device.num” : “10000”,

“device.score” : “4”

},

{ “key” : “quality”,

“device.code” : “D\*”,

“device.num” : “10004”,

“device.score” : “1”

}]

}

}\r\n

## PLC수집 시작 Response 프로토콜(M/W -> 응용서비스)

* **Response JSON 스키마**

{

“jsonrpc” : “2.0”,

“id” : {“type” : “string”},

“result” : {

“protocol.version” : {“type” : “string”},

“event.id” : {“type” : “string”},

“proc.data” : {“type” : “string”}

},

“error” : {

“code” : {“type” : “number”},

“message” : {“type” : “string”}

}

}\r\n

* Response 항목 상세

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **항목명** | | **항목명(한글)** | **필수** | **설명** | **비고** |
| 1 | jsonrpc | | Json rpc 버전 | O | Json rpc 프로토콜 버전 | 2.0 |
| 2 | id | | 통신ID | O | Req-Res 통신 ID |  |
| 3 | result | | 결과정보 | X |  | 장애 발생시 ‘null’ |
| 4 |  | protocol.version | 프로토콜 버전 | O | 프로토콜 버전 |  |
| 5 |  | event.id | 이벤트 ID | O | 명령 처리의 이벤트ID |  |
| 6 |  | proc.data | 처리시간 | O | 명령 처리시간 | yyyymmdd hh:mm:ss.SSS |
| 7 | error | | 에러정보 | X | 장애 발생시 에러정보 | 정상 처리시 ‘null’ |
| 8 |  | code | 에러코드 | O | 에러코드 |  |
| 9 |  | message | 에러메세지 | O | 에러메세지 |  |

* Response JSON 샘플

{

“jsonrpc” : “2.0”,

“id” : “1”,

“result” : {

“protocol.version” : “1.0”,

“event.id” : “event1”,

“proc.data” : “20190801 17:32:54.100”

},

“error” : {

“code” : -32000,

“message” : “PLC connection failed.”

}

}\r\n

# PLC 수집 정지 프로토콜

* 응용서비스와 미들웨어 간 PLC수집 정지 명령을 위한 TCP/IP 기반의 프로토콜
* 전송 데이터는 JSON표준 형식으로 전송
* 전송 문자 타입은 UTF-8 인코딩 사용.
* 전송 종료 코드는 CR LF ('\r\n')를 사용.

## PLC수집 정지 Request 프로토콜(응용서비스 -> M/W)

## PLC수집 정지 Response 프로토콜(M/W -> 응용서비스)

# PLC 수집 조회 프로토콜

* 응용서비스와 미들웨어 간 PLC수집 기동 상태 조회를 위한 TCP/IP 기반의 프로토콜
* 전송 데이터는 JSON표준 형식으로 전송
* 전송 문자 타입은 UTF-8 인코딩 사용.
* 전송 종료 코드는 CR LF ('\r\n')를 사용..

## PLC수집 조회 Request 프로토콜(응용서비스 -> M/W)

## PLC수집 조회 Response 프로토콜(M/W -> 응용서비스)

# PLC 수집 일괄정지 프로토콜

* 응용서비스와 미들웨어 간 PLC수집 일괄 정지 명령을 위한 TCP/IP 기반의 프로토콜
* 전송 데이터는 JSON표준 형식으로 전송
* 전송 문자 타입은 UTF-8 인코딩 사용.
* 전송 종료 코드는 CR LF ('\r\n')를 사용.

## PLC수집 일괄정지 Request 프로토콜(응용서비스 -> M/W)

## PLC수집 일괄정지 Response 프로토콜(M/W -> 응용서비스)

# PLC 수집 정보 Publish 프로토콜

* 응용서비스와 미들웨어 간 PLC수집정보 전달을 위한 MQ(PUB-SUB)기반의 Broadcast용 publish 프로토콜
* 전송 데이터는 JSON표준 형식으로 전송
* 전송 문자 타입은 UTF-8 인코딩 사용.
* 전송 종료 코드는 CR LF ('\r\n')를 사용.

## PLC수집 정보 Publish 프로토콜(M/W -> 응용서비스)

* Publish JSON 스키마

{

“jsonrpc” : “2.0”,

“id” : {“type” : “string”},

“result” : {

“protocol.version” : {“type” : “string”},

“event.id” : {“type” : “string”},

“proc.data” : {“type” : “string”}

“items” : [

{ “key” : {“type” : “string”},

“value” : {“type” : “string”}

}],

}

“error” : {

“code” : {“type” : “number”},

“message” : {“type” : “string”}

}

}\r\n

* Publish 항목 상세

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **항목명** | | | **항목명(한글)** | **필수** | **설명** | **비고** |
| 1 | jsonrpc | | | Json rpc 버전 | O | Json rpc 프로토콜 버전 | 2.0 |
| 2 | id | | | 통신ID | O | Pub 통신 ID | auto increment |
| 3 | result | | | 결과정보 | O |  |  |
| 4 |  | protocol.version | | 프로토콜 버전 | O | 프로토콜 버전 |  |
| 5 |  | event.id | | 이벤트 ID | O | 명령 처리의 이벤트ID |  |
| 6 |  | proc.data | | 처리시간 | O | 명령 처리시간 | yyyymmdd hh:mm:ss.SSS |
|  |  | items | | 아이템정보 | X | 수집아이템 정보 | 장애 발생시 ‘null’ |
|  |  |  | key | 키 | O | 수집 아이템 이름 | 수집시작 명령 시 전달받은 아이템 이름과 동일 |
|  |  |  | value | 값 | O | 수집 아이템 값 | 16진수 HEX값 |
| 7 | error | | | 에러정보 | X | 장애 발생시 에러정보 | 정상 처리시 ‘null’ |
| 8 |  | code | | 에러코드 | O | 에러코드 |  |
| 9 |  | message | | 에러메세지 | O | 에러메세지 |  |

