**EFEM-Tool 인터페이스 사양서**



# 변경이력

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 버전번호 | 0.9 | 날짜 | 2022-10-27 | 변경자 | 박원석 |
| 1. 전체 EFEM 사양서에서 별도 문서로 분리 | | | | | |
| 버전번호 | 1.0 | 날짜 | 2022-10-31 | 변경자 | 박원석 |
| 1. 소켓 메시지 보완  2. 전체 시나리오 추가  3. Multi-Stage PIO 시퀀스 추가 | | | | | |
| 버전번호 | 1.1 | 날짜 | 2022-12-03 | 변경자 | 박원석 |
| 1. CASSETTE\_VERIFY\_REQ를 WORKSEQ\_VERIFY\_REQ로 변경함.  2. PORT READY 상태 등을 EFEM이 소켓통신으로 Tool에게 물어보는 것이 아니라 PIO 시퀀스로 연동되도록 수정함. | | | | | |
| 버전번호 | 1.2 | 날짜 | 2022-12-08 | 변경자 | 박원석 |
| 1. [(내용가기)](#bm_pio_basic_include_shutter_ko)  PIO신호는 셔터 제어 시그널을 기본적으로 포함함.  2. [(내용가기)](#bm_pio_block_separate_ko)  PIO 블럭을 load를 위한 신호와, unload 를 위한 신호를 구분함. 기존의 12선으로 load/unload 혼용하던 것을 load/unload 각각 12선을 사용함으로 싱글인 경우도 24개의 선이 필요함.  3. [(내용가기)](#bm_number_of_socket_ko)  EFEM, Tool 각각 송/수신 용도로 Server Socket/Client Socket을 가져야 한다는 제약을 제거하고, 하나의 소켓으로 송/수신을 사용할 수 있도록 함.  4. [(내용가기)](#bm_메시지포맷을json에통합_ko)  소켓 통신 데이터 포맷을 json 객체 하나에 데이터를 다 넣고, 앞/뒤로 stx/etx를 추가하는 것으로 변경함.  5. [(내용가기)](#bm_using_json_schema_ko)  각 메시지를 설명할때 json schema를 사용하도록 수정함. | | | | | |
| 버전번호 | 1.3 | 날짜 | 2023-01-17 | 변경자 | 박원석 |
| 1. WORKSEQ\_VERIFY\_REQUEST 에서 HANDOVER\_PORT 프로퍼티 삭제.  2. WORKSEQ\_END\_VERIFY\_REQUEST를 추가하여 한 work sequence가 끝났음을 확인한다.  3. PIO-Socket Integration Scenario 를 추가하여 소켓 메시지와 PIO 신호의 전달의 순서를 함께 확인하도록 함. | | | | | |
| 버전번호 | 1.31 | 날짜 | 2023-01-17 | 변경자 | 박원석 |
| 1. Added STAGE\_NO to UNLOAD\_REQUEST\_STAGE\_NO | | | | | |
| 버전번호 | 1.32 | 날짜 | 2023-02-02 | 변경자 | 박원석 |
| 1. GLASS\_DATA\_REPORT\_REQUEST 메시지에서 SUBSTRATE\_ORDER를 SLOT\_NO로 변경함. | | | | | |
| 버전번호 | 1.33 | 날짜 | 2023-02-13 | 변경자 | 박원석 |
| LOAD\_REQUEST\_STAGE\_NO, LOAD\_RESPONSE\_STAGE\_NO에 GLASS\_ID 추가 | | | | | |
| 버전번호 | 1.34 | 날짜 | 2023-02-14 | 변경자 | 박원석 |
| 1. EFEM → Tool WorkOrder 전송을 위한 메시지 추가(WORKORDER\_START\_REQUEST). WorkOrder는 전체 작업의 내용이 들어 있으므로 메시지 필요함.  2. 기존의 WORKSEQ\_VERIFY\_REQUEST를 WORKSEQPP\_START\_REQUEST로 변경함. WORKSEQPP\_START\_REQUEST는 동일한 PP의 글래스작업을 가지고 있음. | | | | | |
| 버전번호 | 1.35 | 날짜 | 2023-02-16 | 변경자 | 박원석 |
| 1. WorkOrder설명과 Manual Mode 전환 작업에 대한 내용 추가 | | | | | |
| 버전번호 | 1.36 | 날짜 | 2023-02-17 | 변경자 | 박원석 |
| 1. WORKORDER\_START\_REQUEST 메시지에 efem의 workseq 번호만 포함되어 있는 것을 workseqpp 번호를 포함하도록 수정함.  2. WORKSEQPP\_START\_REQUEST에 slot별 데이터에 SLOT\_STATE 추가. slot에 실제로 Glass가 존재하는지의 정보. | | | | | |
| 버전번호 | 1.37 | 날짜 | 2023-02-21 | 변경자 | 박원석 |
| 1. PIO 시퀀스 중 예외 상황 발생시 글래스 소유권 및 시퀀스 재시작 주체 추가[Contents](#bm_ownsership_seqrestarter) | | | | | |
| 버전번호 | 1.40 | 날짜 | 2023-02-22 | 변경자 | 박원석 |
| 1. 기존 멀티 스테이지에 있었던 소켓 메시지 일부 삭제.LOAD\_EANBLE\_STAGE, LOAD\_ACCESS\_STAGE  2. PIO 시퀀스의 Timeout 명칭 정의 및 설명.  3. PIO 시퀀스의 상세설명 추가 및 timeout 상황 및 조치 방법 추가[Ref. Contents](#bm_pio_seq_detail_new)  [4. Add socket messages. GLASS\_SCRAP\_NOTIFY\_REQUEST(RESPONSE)](#bm_pio_seq_detail_new)  [Add socket message. EQ\_STATE\_INFO](#bm_pio_seq_detail_new) | | | | | |
| 버전번호 | 1.41 | 날짜 | 2023-02-28 | 변경자 | 박원석 |
| Multi Stage의 타이밍 차트 load/unload 그림 수정.[Ref. Contents](#bm_multistage_timechart_load) | | | | | |
| 버전번호 | 1.42 | 날짜 | 2023-03-13 | 변경자 | 박원석 |
| socket/pio full 시나리오 추가.[Ref. Contents](#bm_pio_socket_integration_scenario) | | | | | |
| 버전번호 | 1.43 | 날짜 | 2023-04-04 | 변경자 | 박원석 |
| 결선 방식에 배선을 EFEM/Tool 각각 제공하는 것으로 명시함.[Ref. Contents](#bm_wiring) | | | | | |
| 버전번호 | 1.44 | 날짜 | 2023-04-06 | 변경자 | 박원석 |
| 1. Timechart의 각 시퀀스에 번호 표시하고 이 번호와 연관하여 시퀀스 설명함.  2. 각 Timeout 발생 했을때 어디서 부터 다시 시작해야 할지를 명시함. (restart stage) | | | | | |
| 버전번호 | 1.45 | 날짜 | 2023-04-07 | 변경자 | 박원석 |
| 1. GLASS\_DATA\_REPORT\_REQUEST 메시지에 CARRIER\_ID, PORT\_ID 추가  2. GLASS\_DATA\_REPORT\_REQUEST 메시지에 MAPPED\_GLASSCOUNT, THIS\_GLASSORDER 추가  3. PIO 타임아웃 발생시 약속된 재시작을 위한 PIO\_RESTART\_REQUEST 추가 | | | | | |
| 버전번호 | 1.46 | 날짜 |  | 변경자 |  |
| 1. WORKORDER\_START\_REQUEST/CJID 추가. ControlJob을 식별하기 위해서 필요함.  2. WORKORDER\_START\_REQUEST/WORKSEQUENCEPP/WORKSEQ\_NO 삭제. 사용처가 없음.  3. WORKORDER\_START\_REQUEST/WORKSEQUENCEPP/PJID 추가. ProcessJob을 식별하기 위해서 필요함.  4. WORKSEQPP\_START\_REQUEST/WORKSEQPP\_DATA/SEQNO 삭제.  5. WORKSEQPP\_START\_REQUEST/WORKSEQPP\_DATA/PJID 추가.  6. WORKSEQPP\_START\_REQUEST/WORKSEQPP\_DATA/CARRIER\_SLOTS/PPID 삭제.  7. WORKORDER\_START\_REQUEST/WORKSEQUENCEPP/WORKSEQPP\_CARRIER\_SLOTS 추가. 매뉴얼 작업일때 cj 생성을 위한 것임.  8. WORKORDER\_START\_REQUEST  WORKSEQPP\_START\_REQUEST | | | | | |
| 버전번호 | 1.47 | 날짜 | 2023-04-27 | 변경자 | 박원석 |
| 1. WORKORDER\_START\_REQUEST 메시지에 작업할 모든 글래스 리스트 포함. WORKSEQPP\_START\_REQUEST에 있던 것을 앞으로 당긴 것임.  2. WORKORDER\_START\_REQUEST, WORKSEQPP\_START\_REQUEST, GLASS\_DATA\_REPORT\_REQUEST 메시지에서 ControlJob, ProcessJob 관련 정보는 제거함. Tool에서는 이 정보를 사용하지 않음.  3. Tool이 더 이상 ProcessJob/ControlJob을 수신하지 않으므로, ProcessJob/ColtrolJob과 WorkOrder/WorkseqPP간의 관계를 설명하는 부분을 삭제함.  4. 매뉴얼 모드에 대한 설명 삭제  5. AutoMode와 Manual Mode 간의 시나리오 차이가 없으므로 동일한 시나리오로 변경함.  6. section #4 -> section #2  7. section #5 -> section #3  8. 소켓 메시지 샘플 모음 추가 | | | | | |
| 버전번호 | 1.48 | 날짜 | 2023-05-12 | 변경자 | 박원석 |
| 1. 소켓 메시지의 optional 항목도 내용이 비어 있더라도 항목은 존재해야 한다고 명시함.  2. EQP\_STATE\_INFO 메시지의 용도를 강화함. | | | | | |
| 버전번호 | 1.49 | 날짜 | 2023-05-26 | 변경자 | 박원석 |
| 1. GLASS\_DATA\_REPORT\_REQUEST에 PPID 추가.  degas 설비같이 work정보를 직접 받지 않는 설비에는 글래스 한 장당 현재 가공하는 ppid를 알 필요가 있음. | | | | | |
| 버전번호 | 1.50 | 날짜 | 2023-06-14 | 변경자 | 박원석 |
| 1. Exchagne 시퀀스를 언로딩중 기본 동작으로 함. (§2.2.3)  2. PIO 타임아웃 발생 시, 에러를 발생하고 작업자 조치하며, 글래스 소유권에 따라서 재시작을 수행하도록 수정함.(§2.2.2)  이로 인한 PIO\_RESTART\_REQUEST 메시지 삭제  3. EFEM의 작업완료 처리를 위한 GLASS\_SCRAP\_NOTIFY\_REQUEST, GLASS\_DATA\_REPORT\_REQUEST 메시지 중요 내용 명시 (§2.3.2.2.1)  4. WORKORDERSTART\_REQUEST/IsManual 속성 추가. 이것으로 Auto/Manual Mode 구분함.(§2.3.2.2.3.1)  MANUAL\_START\_REQUEST/MANUAL\_START\_RESPONSE 메시지 삭제  5. GLASS\_SCRAP\_NOTIFY\_RESPONSE 메시지 추가  6. MultiStage pio 타임 차트에서 뒷 부분의 LOAD\_REQUEST\_STAGE\_NO (stageno=-1)을 보내서 slot 번호를 무효화 한다는 단계 제거함.  7. 각 타임차트의 Pre Condtion 컬럼을 복구함. 이후에도 이슈가 될 경우에 참조하기 위함임.  8. auto/emo\_st 비트 사용방법 추가(§2.2.4)  9. 현장에서 PIO 케이블, 랜 케이블을 한화에서 제공하는 것을 명시함. (테스트 할때는 각자 준비)(§2.2.5) | | | | | |
| 버전번호 | 1.60 | 날짜 | 2023-08-17 | 변경자 | 박원석 |
| 1. 속성의 필요 여부를 명확히 하기 위해서 모든 메시지의 object에 required 추가함.  2. WORKSEQPP\_START\_REQUEST 메시지에 PJID 속성 추가. 이것은 해당 작업의 EAP로 부터 수신한 Process Job Id 임. ($2.5.2.2.3.3)  3. PIO 타임아웃 발생 시 글래스 소유권이 상대편으로 넘어갔을 경우 현재 글래스에 대한 시퀀스를 종료하도록 명시함.(§2.4.2.2.3)  4. EAP가 CJAbort 명령을 EFEM에 내렸을때 Abort 동작을 하기 위한 소켓 메시지 추가. (§2.3.1)  5. GLASS\_REPORT\_REQUEST 소켓 메시지에 MANUAL\_INSP 속성 누락된 것 복원(§2.5.2.2.3.9)  6. WORKSEQPP\_START\_RESPONSE의 Ack가 -2인 경우를 추가함. Tool의 internal error 상태($2.5.2.2.3.4) | | | | | |
| 버전번호 | 1.61 | 날짜 | 2023-08-31 | 변경자 | 박원석 |
| 1. EAP의 CJAbort 명령을 수행하기 위한 소켓 메시지 6개 추가. (§2.5.2.2.3.14 ~ 19)  2. CJAbort를 수행하는 시퀀스 그림 추가.(§2.3.1.1, §2.3.1.2)  3. GLASS\_DATA\_REPORT\_REQUEST, GLASS\_SCRAP\_NOTIFY\_REQUEST의 ULGFlag 는 특별한 설비인 “unpacking loader” 에만 해당함을 명시함. (§2.5.2.2.1) | | | | | |
| 버전번호 | 1.62 | 날짜 | 2023-09-01 | 변경자 | 박원석 |
| 1. EAP가 시작한 Abort와 Tool에서 시작한 Abort를 구분해서 설명함.(§2.3.1.1 ~ 2)  2. Abort를 진행하기 위한 EFEM-Tool간의 소켓 메시지 이름 변경 (§2.5.2.2.3.14 ~ 19)  3. Tool의 모드전환 설명을 삭제하고, WORKORDER\_START\_REQUEST 메시지의 IsManual 속성의 사용용도를 설명함. (§2.1)  4. GLASS\_DATA\_REPORT\_REQUEST에 FINISHED\_SUBSTEPS 속성 추가. (§2.5.2.2.3.9)  5. "PIO Items for Multi Stage Unload EFEM" item을 아래와 같이 정정함.(§2.4.7.2.1)  pin3: reserved1→exchange\_ready  pin4: robot\_busy  pin6: reserved2→eload\_complete  pin9: reserved3  pin10: reserved4  pin11: reserved5  터미널 블럭 그림에 반영함.(§2.4.7.2)  타임차트에 반영함.(§2.4.7.2.3.1, §2.4.7.2.3.2) | | | | | |

# 1 EFEM-Tool-Host 연결모델

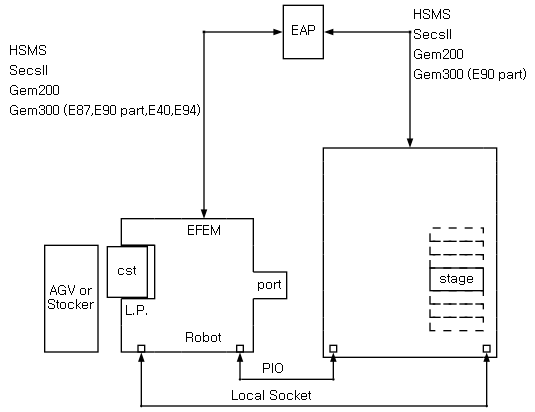


Figure 1 Connection Type ( EFEM - Tool - Host)

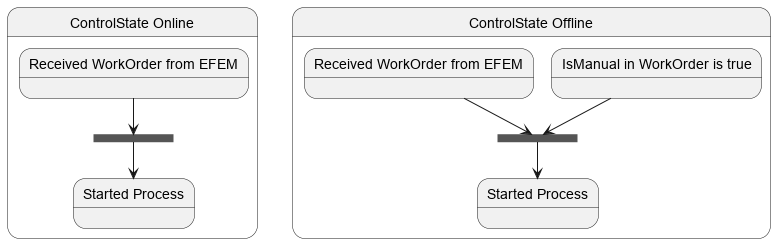
# 2 EFEM - Tool 통신 사양

## 2.1 Tool의 작업시작 조건

Both(In/Out) type EFEM이 연결된 Tool은 EFEM이 전달하는 WorkOrder 정보만 가지고 작업할 수 있다.

다만 Tool의 Control Stage가 Offline일때는 WorkOrder의 IsManual 속성이 true일때만 작업을 시작할 수 있다.

WorkOrder의 IsManual 속성이 true 인 것은 이 WorkOrder가 EAP에게서 받은 것이 아니라, 사용자가 수동으로 만든 것이며 상위 연결 없이도 작업을 하려는 의도로 만든 것임을 의미한다.



## 2.2 전체 시나리오

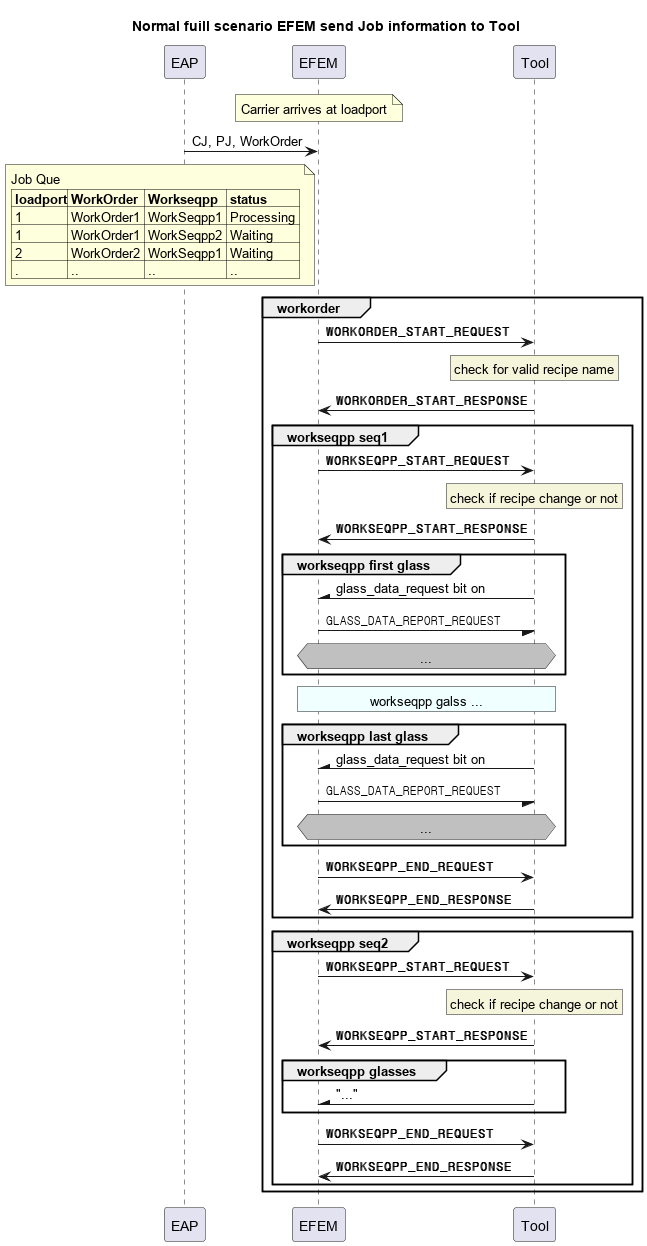


Figure 2 Normal Scenario

| EAP |  | EFEM |  | Tool |
| --- | --- | --- | --- | --- |
|  | ← | S6F11 Ready To Load |  |  |
|  | ← | S6F11 Carrier Arrived |  |  |
|  | ← | S6F11 Carrier Clamped |  |  |
|  | ← | S6F11 Carrier ID Read |  |  |
| S3F17 Carrier Action Request (ProceedWithCarrier) | → |  |  |  |
|  | ← | S3F18 Acknowledge |  |  |
|  | ← | S6F11 Slot Map Report |  |  |
| S16F15 Process Job Create Request | → |  |  |  |
|  | ← | S16F16 Acknowledge |  |  |
| S14F9 Create Control Job | → |  |  |  |
|  | ← | S14F10 Acknowledge |  |  |
| S3F41 WorkOrder(CMS) | → |  |  |  |
|  |  | WORKORDER\_START\_REQUEST | → |  |
|  |  |  |  | Verify recipe name |
|  |  |  | ← | WORKORDER\_START\_RESPONSE |
| First workseqpp starts |  |  |  |  |
|  |  | WORKSEQPP\_START\_REQUEST | → |  |
|  |  |  |  | check recipe change needed |
|  |  |  | ← | WORKSEQPP\_START\_RESPONSE |
|  |  | Glass Loading sequence (PIO + socket) |  |  |
|  |  |  |  | Glass Processing (Report events to Host) |
|  |  | Glass Unlading sequence (PIO + socket) |  |  |
|  |  | WORKSEQPP\_END\_REQUEST | → |  |
|  |  |  | ← | WORKSEQPP\_END\_RESPONSE |
| First workseqpp end |  |  |  |  |
|  | ← | S6F11 Process Job Completed |  |  |
| next workseqpp starts |  |  |  |  |
| glasses in workseqpp process |  |  |  |  |
| next workseqpp ends |  |  |  |  |
|  | ← | S6F11 Process Job Completed |  |  |
| last workseqpp starts |  |  |  |  |
| glasses in workseqpp process |  |  |  |  |
| last workseqpp ends |  |  |  |  |
|  | ← | S6F11 Process Job Completed |  |  |
|  | ← | S6F11 Control Job Completed |  |  |
|  | ← | S6F11 Ready To Unload |  |  |
|  | ← | S6F11 Carrier Removed |  |  |
|  | ← | S6F11 Ready To Load |  |  |

## 2.3 예외상황 처리

### 2.3.1 Abort

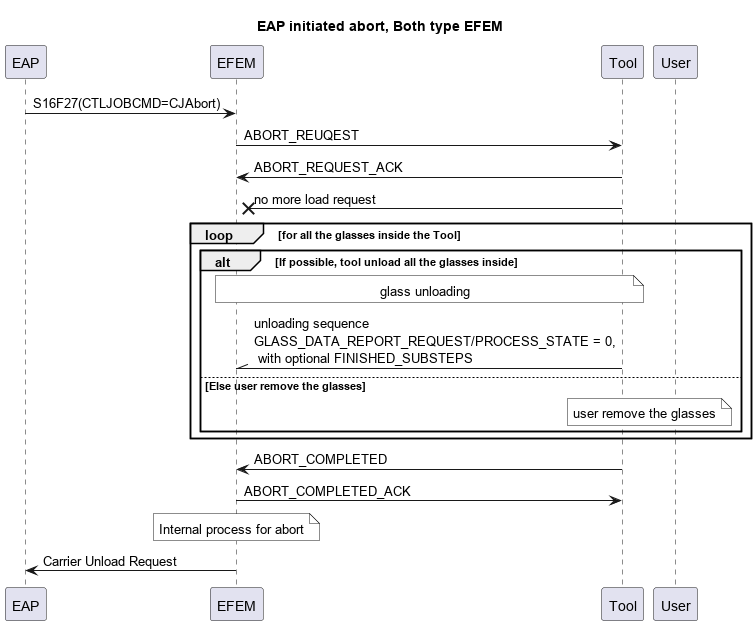
EAP의 명령에 의해서든 Tool에서 이상 상황이 발생해서든 Abort는 글래스 공정을 중지하고 글래스들은 Tool에서 EFEM으로 언로딩되며 EFEM은 캐리어를 배출해야 한다.

기존에 있는 WORKSEQPP\_END\_REQUEST 메시지는 workseqpp가 정상 종료됨을 알리는 것이므로 Abort 처리를 할 수 있는 추가 메시지가 필요하다.

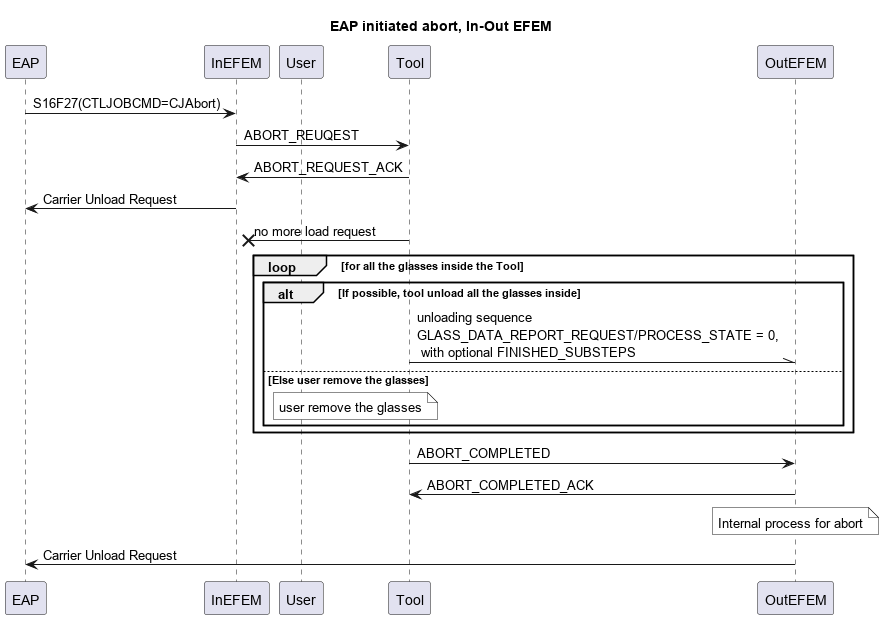
Abort 동작을 하기 위해서 ABORT\_REQUEST(EFEM → Tool), ABORT\_COMPLETED(EFEM ← Tool), ABORT\_STARTED(EFEM ← Tool) 메시지를 추가한다.

#### 2.3.1.1 EAP 명령에 의한 Abort

##### 2.3.1.1.1 Both type EFEM

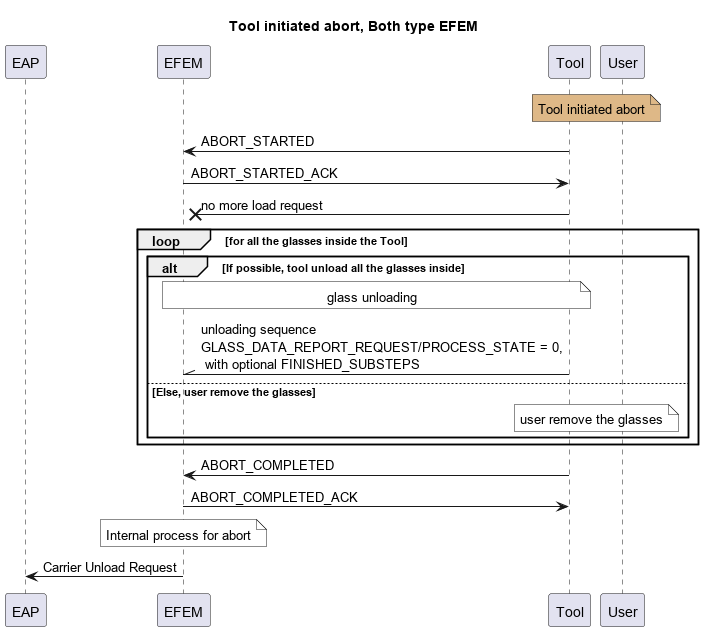


##### 2.3.1.1.2 In-Out type EFEM

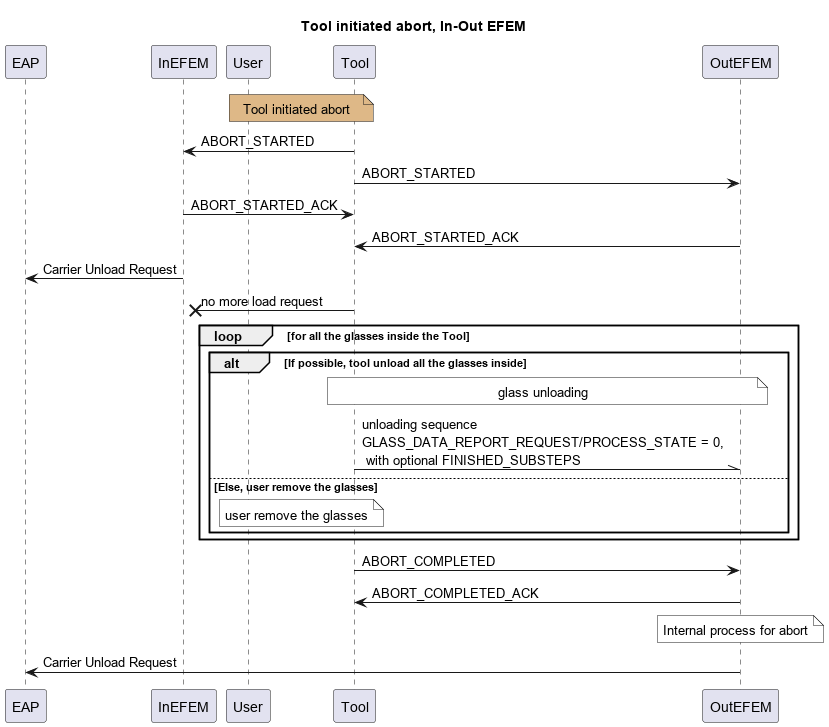


#### 2.3.1.2 Tool에서 시작한 Abort

##### 2.3.1.2.1 Both type EFEM



##### 2.3.1.2.2 In-Out type EFEM



## 2.4 PIO 통신

PIO 구성은 EFEM-Tool 간의 여러 케이스를 다 적용할 수 있도록 구성한다.

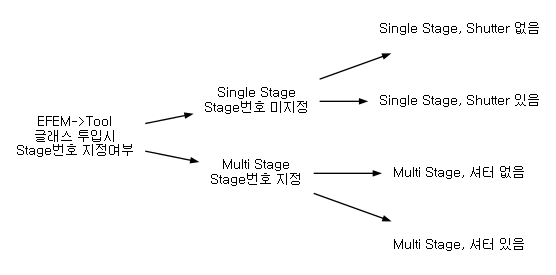


Figure 3 PIO 구성 트리

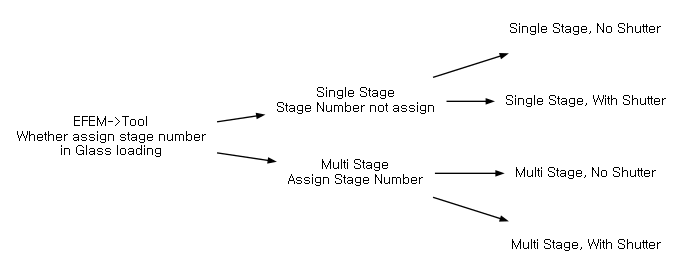


Figure 4 PIO 구성 트리

### 2.4.1 PIO 시퀀스 중 글래스 소유권 이전 시점

| Load/Unload | Glass 소유권 이전 | 시점 |
| --- | --- | --- |
| Loading | EFEM → Tool | EFEM이 load\_complete bit을 ON 한 후.  Tool이 load\_complete\_confirm을 ON하지 않았다 하더라도 글래스의 소유권은 Tool에 있다.  이후 타임아웃 조치는 글래스가 Tool에 있다는 가정하에 진행한다. |
| Unloading | EFEM ← Tool | EFEM이 unload\_complete bit을 ON 한 후.  Tool이 unload\_complete\_confirm을 ON하지 않았다 하더라도 글래스의 소유권은 EFEM에 있다.  이후 타임아웃 조치는 글래스가 EFEM에 있다는 가정하에 진행한다. |

### 2.4.2 PIO 타임아웃 처리방법

pio timeout 발생 시 타임 발생시 에러를 발생하고 사용자로 하여금 조치하게 해야 한다.

#### 2.4.2.1 Timeout 명칭

타임아웃은 load\_request\_timeout 와 같이 발생한 시점으로 명칭하고 타임차트에서는 아래와 같은 기호로 명시한다.

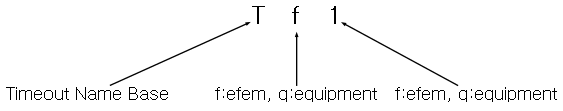


Figure 5 Timeout 명칭 설명

#### 2.4.2.2 PIO 타임 아웃처리를 위해서 필요한 기능

##### 2.4.2.2.1 글래스 생성/삭제 기능

"글래스 생성/삭제 기능" 이란 에러 발생 후, 실물 조치 후 실물과 데이터를 맞추기 위해서 사용자가 UI를 통해서 Tool 혹은 EFEM에서 글래스 정보를 생성 혹은 삭제 할 수 있는 기능을 뜻한다.

| EFEM/Tool | Necessary Function | 상황 |
| --- | --- | --- |
| Tool | 글래스 생성 기능 | EFEM에서 load\_complete\_confirm timeout 발생시.  실제로는 글래스가 Tool에 전달 되었지만 Tool 내부의 오류로 인해서 글래스의 존재를 파악하지 못하는 경우에 글래스 생성 기능이 필요함.  Tool에서 글래스 감지 센서가 있는 경우에도 명시적으로 사용자가 글래스 정보를 확인하는 과정이 필요함. |
| Tool | 글래스 삭제 기능 | EFEM에서 unload\_complete\_confirm timeout 발생시  실제로는 글래스가 Tool에서 제거되었지만 Tool 내부의 오류로 인해서 글래스가 제거되었음을 알지 못하는 경우. 사용자에 의한 글래스 삭제 기능 필요함. |
| EFEM | 글래스 생성/삭제 기능 | EFEM 로봇은 항상 실물 글래스 기반으로 동작해야 하므로 이 기능이 필요없음.. |

##### 2.4.2.2.2 PIO 시퀀스 재시작 기능

"PIO 시퀀스 재시작 기능"을 위한 UI는 기존 "Run" 버튼 등으로 대체가능하다.

| 구분 | "PIO 재시작" 기능의 내용 |
| --- | --- |
| 글래스의 소유권이 이전되지 않은 상황에서 timeout 발생 | 사용자의 에러 조치 후 "PIO 시퀀스 재 시작"을 수행하면 로딩/언로딩의 시퀀스를 처음부터 다시 시작  "PIO 시퀀스 재시작 기능"을 수행하는 경우 EFEM/Tool모두 기존의 이미 수신했던 GLASS\_DATA\_REPORT\_REQUEST 메시지를 다시 수신하더라도 지금 진행하고 있는 글래스의 데이터 임을 인지하고 정상처리 할 수 있어야 한다. |
| 글래스의 소유권이 이전된 후 timeout 발생 | 사용자의 에러 조치 후 "PIO 시퀀스 재 시작"을 수행하면 로딩/언로딩 시퀀스를 마무리한다. |

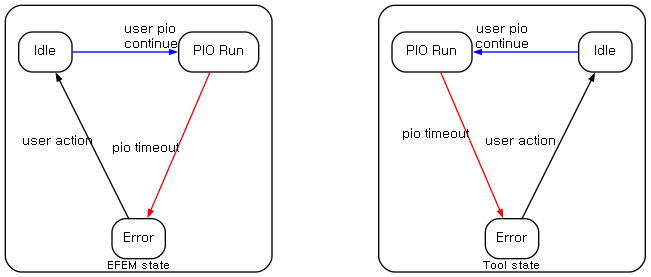


Figure 1 PIO 타임아웃 조치 후 재시작에 따른 설비상태 변화

##### 2.4.2.2.3 글래스 소유권에 따른 PIO 타임아웃 발생시 PIO 재시작 방법

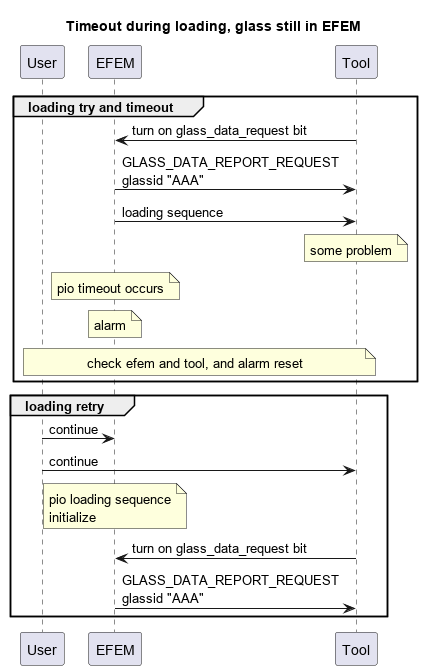


Figure 1 Timeout during Loading, glass still in EFEM

| PIO 타임아웃 발생시 글래스 소유권 | 글래스 소유권에 따른 PIO 재시작 방법 |
| --- | --- |
| Loading, glass in EFEM | 1. 사용자는 에러 조치 후 "PIO 시퀀스 재시작"을 수행한다.  2. Tool은 glass\_data\_request bit를 on하는 것 부터 다시 시작한다.  3. EFEM은 Tool의 glass\_data\_request bit ON을 확인하고 GLASS\_DATA\_REPORT\_REQUEST를 Tool에게 전송하는 것 부터 시작한다. |
| Loading, glass in Tool | 1. 사용자는 에러 조치 및 필요하면 Tool에서 "글래스 생성 기능"을 수행한다.  2. EFEM과 Tool은 이번 글래스에 대한 시퀀스를 각자 마무리하고 다음 글래스를 위한 작업 대기 상태로 진입한다.  3. 사용자는 "EFEM-PIO 시퀀스 재시작"을 통해서 다음 작업을 시작한다. |
| Unloading, glass in Tool | 1. 사용자는 에러 조치 후 "PIO 시퀀스 재시작 기능"을 수행한다.  2. Tool은 GLASS\_DATA\_REPORT\_REQUEST 전송 하는 것 부터 다시 시작한다.  3. EFEM은 Tool의 GLASS\_DATA\_REPORT\_REQUEST를 수신하고 언로딩 시퀀스를 처음부터 진행한다. |
| Unloading, glass in EFEM | 1. 사용자는 에러 조치 및 필요하면 Tool에서 "글래스 삭제 기능"을 수행한다.  2. EFEM과 Tool은 이번 글래스에 대한 시퀀스를 각자 마무리하고 다음 글래스를 위한 작업 대기 상태로 진입한다.  3. 사용자는 "EFEM-PIO 시퀀스 재시작"을 통해서 다음 작업을 시작한다. |

### 2.4.3 언로딩 중 기본 exchange loading(eloading) 진행

공정 설비를 효율적으로 운영하기 위해서 글래스 언로딩 후 바로 로딩을 하는 시퀀스를 기본으로 한다.

Exchange 제어를 위해서 기존의 reserved bit중 필요한 bit를 할당한다.

| 설비/EFEM | 추가 할당하는 bit |
| --- | --- |
| 설비쪽 | 1. exchange\_enable 의 용도로 reserved1을 할당한다.  2. eload\_complete\_confirm 용도로 reserved2를 할당한다. |
| EFEM쪽 | 1. exchange\_ready 용도로 reserved1을 할당한다.  2. eload\_complete 용도로 reserved2를 할당한다. |

Unload exchange 기본 순서는 다음과 같다.

1. EFEM은 unload시 exchange 가능한 상황에서 exchange\_ready on 한다.

2. 설비는 exchange\_enable on 한다. 설비가 exchagne\_enable on 하지 않으면 EFEM은 exchange 를 위한 로딩 글래스를 수취하지 않고 일반 언로딩 동작을 한다.

3. exchange 상황에서 EFEM은 언로딩 후 GLASS\_DATA\_REPORT\_REQUEST 를 설비에게 보내 로딩할 글래스의 정보를 전달한다. 설비는 GLASS\_DATA\_REPORT\_RESPONSE로 회신한다.

4. EFEM은 글래스를 로딩한 후 eload\_complete on 한다.

설비는 eload\_complete\_confirm on 한다.

#### 2.4.3.1 Exchange 로딩할 개구부(포트) 할당

개구부가 하나인 경우 동일한 개구부로 로딩 한다.



Figure 8 개구부가 하나인 경우의 exchange loading개구부가 두 개이고 각각 로딩/언로딩을 할 수 있는 경우 언로딩한 곳으로 로딩한다.

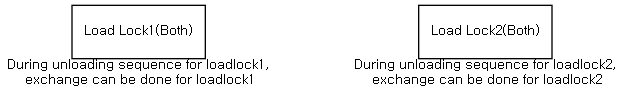


Figure 9 각각 로딩/언로딩을 할 수 있는 두 개의 개구부가 있는 경우Multistage의 경우 하나의 셔터로 두 개의 stage를 연동하는 경우 로딩할 stage no.를 지정해야 한다.

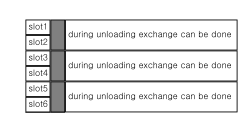


Figure 10 Multistage개구부가 두 개이고 각 개구부가 로딩 혹은 언로딩만 할 수 있는 경우, 각각 개구부별 로딩/언로딩 시퀀스를 진행한다.



Figure 11 각각 로딩/언로딩만 하는 독립적인 두 개의 개구부

### 2.4.4 Auto bit와 emo bit 사용방법

| bit | 사용방법 |
| --- | --- |
| auto | PIO 시퀀스를 진행하기 위해서 항상 ON이 되어야 한다.  아래의 타임차트에서 Precondition 항목에 auto on이 명시적으로 있지 않더라도 기본적으로 auto on을 체크해야 한다  auto 가 off 상태이면 이후 모든 pio 시퀀스는 진행하지 않는다  시퀀스 동작 중 auto off 되면 이후 시퀀스 진행하지 않고 timeout 발생 이후 timeout 처리 방법으로 처리한다... |
| emo\_st | 상태 설비가 emergency 상태에 있으면 on이 된다.  emo\_st on 상태이면 이후 모든 pio 시퀀스는 진행하지 않는다  시퀀스 동작 중 emo\_st on 되면 이후 시퀀스 진행하지 않고 timeout 발생 이후 timeout 처리 방법으로 처리한다.. |

### 2.4.5 결선방식

1. 24C Cable 포설을 통한 Relay 결선 방식

2. EFEM 24C Cable 1가닥 / 공정설비 24C Cable 1가닥 필요

3. EFEM 4점 소형릴레이 6EA 필요 / 공정설비 4점 소형릴레이 6EA 필요

4. Connection Box는 Port 개구부 센터기준 3~5m이내 설치

5. 현장에서 결선할 PIO 케이블은 한화에서 제공합니다.

6. 현장에서 연결할 랜 케이블은 한화에서 제공합니다.

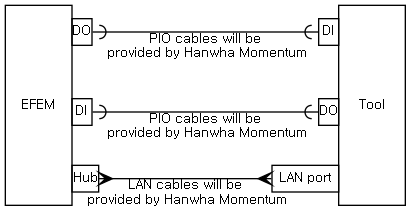


Figure 12 PIO cables

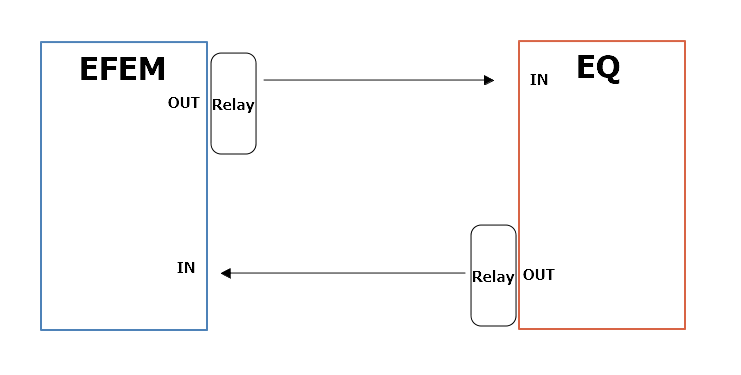


Figure 13 Necessary Relay side

※ EFEM의 terminal box의 정확한 위치는 앱솔릭스와 한화모멘텀 담당자에게 문의하시기 바랍니다.

앱솔릭스 담당자 : 지민호 매니저(minho@Absolcs.com)

한화 모멘텀 담당자 : 하창하 대리(changha.ha@hanwha.com)

Recommend specification (R4T-G6D)

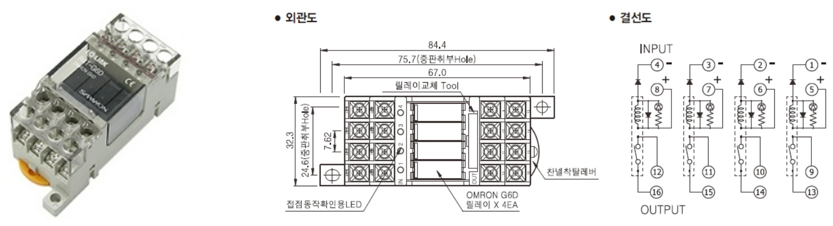


Figure 14 Recommended Relay Connection

### 2.4.6 PIO for Single Stage

일반적으로 stage 번호 지정하지 않고 글래스를 하나 투입하는 경우이며 Shutter는 존재할 수 있다.

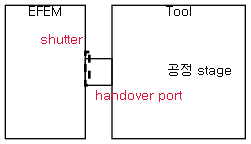


Figure 15 가장 일반적인 single type

글래스 투입하는 곳이 두 군데지만 stage 번호를 지정하지 않기 때문에 아래와 같은 경우도 Single 타입이며, Single 타입의 PIO 셋트를 두 개 써서 대응해야 한다.

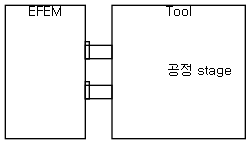


Figure 16 투입구가 두개지만 single type 적용하는 경우

Warning

기존의 12선으로 load/unload 같이 사용 -> load을 위한 12선, unload를 위한 12선 총 24선으로 변경됨.

셔터 개폐를 위한 신호를 기본적으로 할당함. 이 신호는 셔터가 있는 경우 사용하게 되며, 시퀀스가 셔터 유무에 따라서 구분됨.

PIO 시그널은 아래와 같이 load의 경우에 사용하는 것들과 unload의 경우에 사용하는 것들을 분리한다.

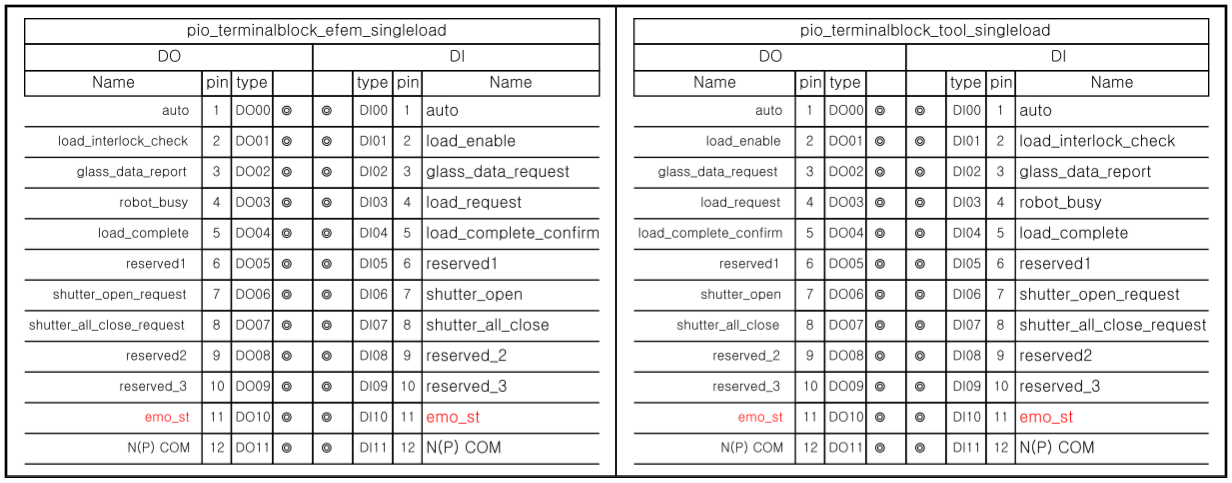


Figure 17 PIO terminal block Single Stage Load

#### 2.4.6.1 PIO for Single Stage Load

##### 2.4.6.1.1 PIO Items for Single Stage Load EFEM

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | 기본적으로 EFEM Auto bit ON 상태일때만 PIO 시퀀스를 진행한다. |
| 2 | load\_interlock\_check | 기본 ON상태. Robot은 이 신호가 OFF일때만 핸드오프충돌영역으로 진입한다. Robot이 핸드오프충돌영역을 벗어나면 다시 ON한다. |
| 3 | glass\_data\_report | EFEM이 GLASS\_REPORT\_REQUEST를 전송한 직후 이 bit를 On 함으로 소켓 데이터가 유효함을 나타낸다. Tool이 load\_complete\_confirm bit를 ON하여 load 동작이 모두 완료되면 EFEM은 이 bit를 OFF한다. |
| 4 | robot\_busy | 로봇이 핸드오프충돌영역으로 진입하기 시작하면 On 합니다. 로봇 위치가 핸드오프 충돌 영역 밖으로 완전히 이동했을때 Off 합니다. |
| 5 | load\_complete | 반송 완료 신호. ROBOT이 Vacuum Off 하여 글래스를 내려 놓았을때 ON 합니다. |
| 6 | reserved1 |  |
| 7 | shutter\_open\_request | 하류 설비에 Shutter Open 요청할 경우 ON 한다. Robot이 동작 중일 경우 상시 ON 한다. |
| 8 | shutter\_all\_close\_request | 하류 설비에 Shutter Close를 요청할 경우 ON 한다.Robot 동작이 완료 될 경우 ON 한다. |
| 9 | reserved2 | exchange 동작시 load\_complete 용도. |
| 10 | reserved3 | Interlock 신호 추가용For adding interlock signal |
| 11 | emo\_st | EFEM Emergency |
| 12 | N(P) Common |  |

##### 2.4.6.1.2 PIO Items for Single Stage Load Tool

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | 기본적으로 Tool Auto bit ON 상태일때만 PIO 시퀀스를 진행한다. |
| 2 | load\_enable | Load 동작 가능 신호  - Interlock 확인 후 Load 동작이 가능 할 경우 ON 한다. |
| 3 | glass\_data\_request | Glass Data 요구신호  - Robot에 Glass Data를 요구 할 경우 ON한다. |
| 4 | load\_request | 반송 시작신호  - Interlock 및 Glass Data 확인 후 Robot 반송 시작 신호 |
| 5 | load\_complete\_confirm | 반송 완료 확인 신호  -. Robot 동작 완료 신호 확인 후 응답 한다. |
| 6 | reserved1 |  |
| 7 | shutter\_open | Shutter Open 신호  -' Shutter가 Open일 경우 ON 한다 |
| 8 | shutter\_all\_close | Shutter Close 신호  -' Shutter가 Close일 경우 ON 한다 |
| 9 | reserved\_2 | Interlock 신호 추가용 |
| 10 | reserved3 | Interlock 신호 추가용 |
| 11 | emo\_st | EQ Emergency |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.6.1.3 Timing chart for Single Stage Load without shutter

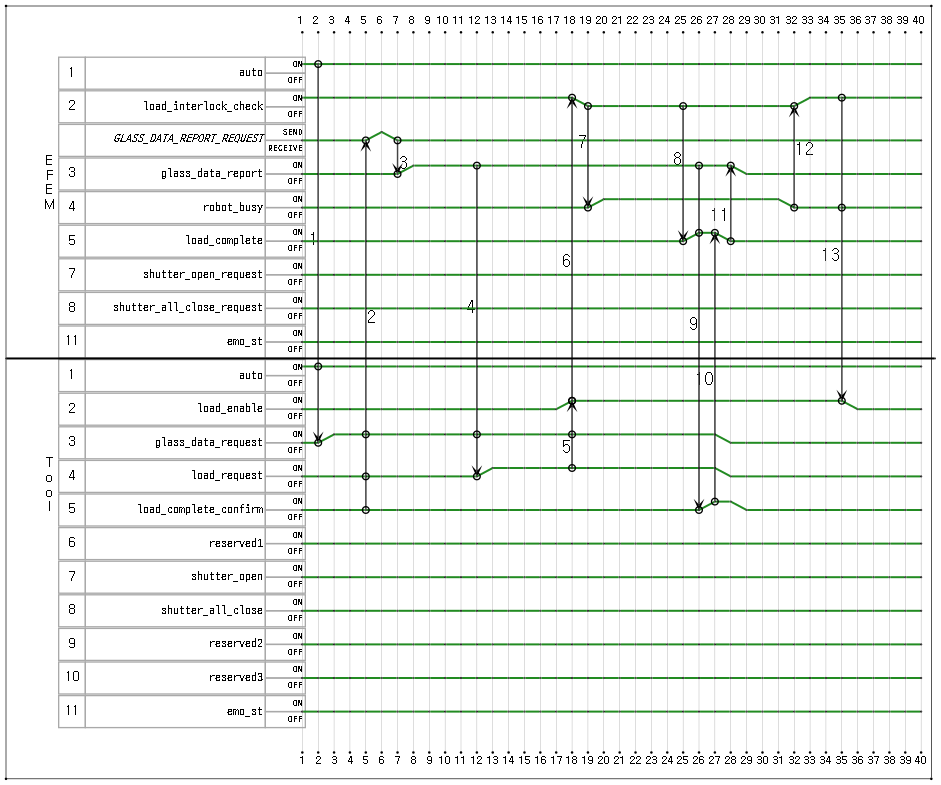


Figure 18 셔터가 없는 Single Stage의 Load 타임 차트

###### 2.4.6.1.3.1 타임 차트 시퀀스 설명

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool glass\_data\_request bit On | efem's auto: On  eq's auto: On | 로딩 시퀀스의 시작은 eq의 glass\_data\_request bit on으로 시작한다. |
| 2 | EFEM GLASS\_DATA\_REPORT\_REQUEST message Send | eq's glass\_data\_request: On  eq's load\_request: Off  eq's load\_complete\_confirm: Off |  |
| 3 | EFEM glass\_data\_report bit On | efem's GLASS\_DATA\_REPORT\_REQUEST: Send |  |
| 4 | Tool load\_request bit On | efem's glass\_data\_report: On  eq's glass\_data\_request: On | eq의 load\_request는 efem의 load\_glass\_data의 유효성을 체크했으므로 글래스 로딩을 시작하자는 의미이다 |
| 5 | Tool load\_enable bit On | eq's load\_request: On  eq's glass\_data\_request: On | eq는 hand off conflict 영역에 글래스를 반입할 조건이 되면 load\_enable 한다. |
| 6 | EFEM load\_interlock\_check bit Off | eq's load\_enable: On | efem은 사전에 글래스를 카세트에서 꺼내지만 handoff conflict 영역으로 진입하지 않고 있다가 이 시점에서는 handoff conflict 영역으로 진입할 준비를 한다. |
| 7 | EFEM robot\_busy bit On | efem's load\_interlock\_check: Off | 글래스를 든 로봇이 busy bit on 하고 hand off conflict 영역으로 진입한다 |
| 8 | EFEM load\_complete bit On | efem's load\_interlock\_check: Off | 로봇이 글래스를 완전히 내려 놓고 나왔다 |
| 9 | Tool load\_complete\_confirm bit On | efem's glass\_data\_report: On  efem's load\_complete: On | 글래스를 감지할 수 있는 Tool은 글래스 감지 센서를 한 번 더 체크하고 load\_complete\_confirm을 on한다. 글래스 감지할 수 없는 Tool은 efem의 load\_complete에 대한 확인으로 load\_complete\_confirm on한다. load\_complete\_confirm on 으로 Tool은 글래스를 소유하고 글래스와 연관된 데이터를 생성했음을 확인하는 것이다. |
| 10 | EFEM load\_complete bit Off | eq's load\_complete\_confirm: On | efem이 tool의 load\_complete\_confirm 이 on 되는 것을 확인하고 load\_complete bit를 off 하는 이유는 Tool에 글래스 전달이 확실히 되는 것을 확인하기 위한 것이다. |
| 11 | EFEM glass\_data\_report bit Off | efem's load\_complete: Off | efem은 load\_complete가 완전히 off된 것을 확인하고 glass\_data\_report를 off 한다 |
| 12 | EFEM load\_interlock\_check bit On | efem's robot\_busy: Off | efem은 robot 이 로딩 영역을 빠져 나온 후 robot busy를 off 하면서 load\_interlock\_check를 on한다 |
| 13 | Tool load\_enable bit Off | efem's load\_interlock\_check: On  efem's robot\_busy: Off |  |

##### 2.4.6.1.4 Timing chart for Single Stage Load with shutter

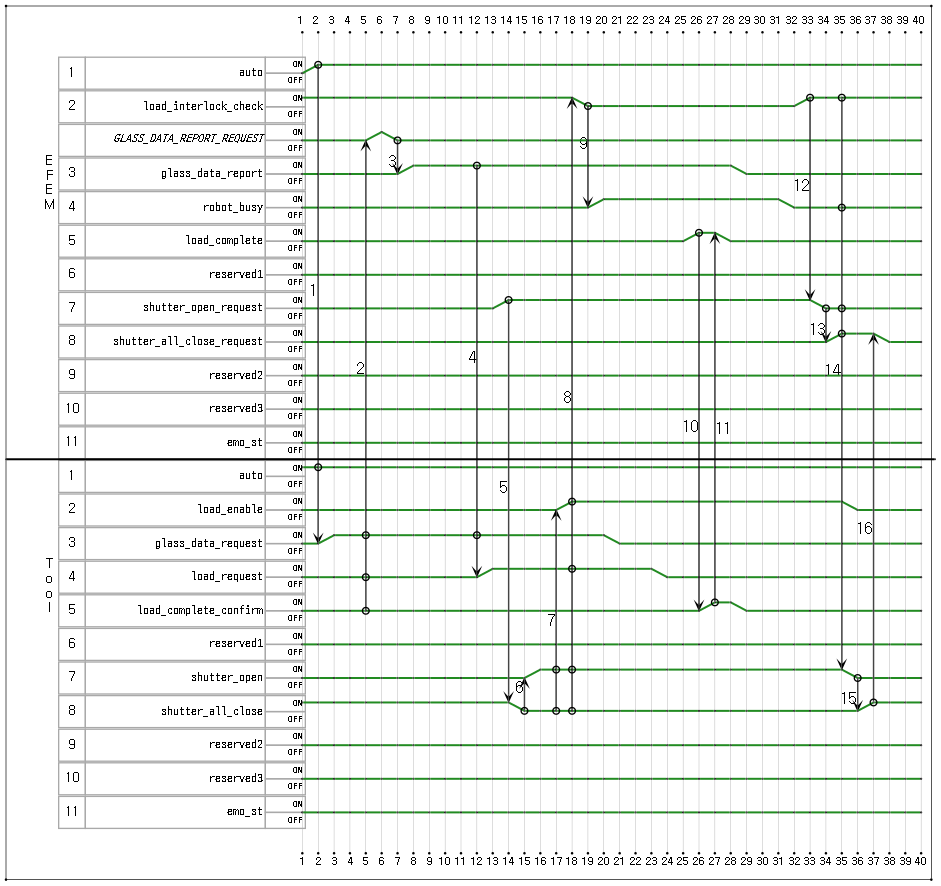


Figure 19 셔터가 있는 Single Stage의 Load 타임 차트

###### 2.4.6.1.4.1 타임 차트 시퀀스 설명

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool glass\_data\_request bit On | efem's auto: On  eq's auto: On | 로딩 시퀀스의 시작은 eq의 glass\_data\_request bit on으로 시작한다. |
| 2 | EFEM GLASS\_DATA\_REPORT\_REQUEST message Send | eq's load\_complete\_confirm: Off  eq's load\_request: Off  eq's glass\_data\_request: On |  |
| 3 | EFEM glass\_data\_report bit On | efem's GLASS\_DATA\_REPORT\_REQUEST: Send |  |
| 4 | Tool load\_request bit On | efem's glass\_data\_report: On  eq's glass\_data\_request: On | eq의 load\_request는 efem의 load\_glass\_data의 유효성을 체크했으므로 글래스 로딩을 시작하자는 의미이다 |
| 5 | Tool shutter\_all\_close bit Off | efem's shutter\_open\_request: On |  |
| 6 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off | 셔터를 열때 eq는 efem의 shutter\_open\_request bit를 직접적으로 보고 움직이는 것이 아니라 내부의 shutter\_all\_close bit의 꺼짐을 보고 shutter open 동작을 한 후 shutter\_open bit를 켜게 된다 |
| 7 | Tool load\_enable bit On | eq's shutter\_open: On  eq's shutter\_all\_close: Off |  |
| 8 | EFEM load\_interlock\_check bit Off | eq's shutter\_all\_close: Off  eq's shutter\_open: On  eq's load\_request: On  eq's load\_enable: On | efem은 사전에 글래스를 카세트에서 꺼내지만 handoff conflict 영역으로 진입하지 않고 있다가 이 시점에서는 handoff conflict 영역으로 진입할 준비를 한다. 셔터가 열렸는지 확인하는 항목만 하나 더 추가된 것이다. |
| 9 | EFEM robot\_busy bit On | efem's load\_interlock\_check: Off | 글래스를 든 로봇이 진입하면서 busy bit를 on한다 |
| 10 | Tool load\_complete\_confirm bit On | efem's load\_complete: On | 글래스를 감지할 수 있는 Tool은 글래스 감지 센서를 한 번 더 체크하고 load\_complete\_confirm을 on한다. 글래스 감지할 수 없는 Tool은 efem의 load\_complete에 대한 확인으로 load\_complete\_confirm on한다. load\_complete\_confirm on 으로 Tool은 글래스를 소유하고 글래스와 연관된 데이터를 생성했음을 확인하는 것이다. |
| 11 | EFEM load\_complete bit Off | eq's load\_complete\_confirm: On | efem은 load\_complete\_comfirm이 on되는 것을 확인하고 load\_complete를 off한다 |
| 12 | EFEM shutter\_open\_request bit Off | efem's load\_interlock\_check: On | efem은 홈 포지션으로 온 이후 shutter\_open\_request bit를 off한다 |
| 13 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off |  |
| 14 | Tool shutter\_open bit Off | efem's robot\_busy: Off  efem's load\_interlock\_check: On  efem's shutter\_open\_request: Off  efem's shutter\_all\_close\_request: On |  |
| 15 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off |  |
| 16 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On |  |

#### 2.4.6.2 PIO for Single Stage Unload

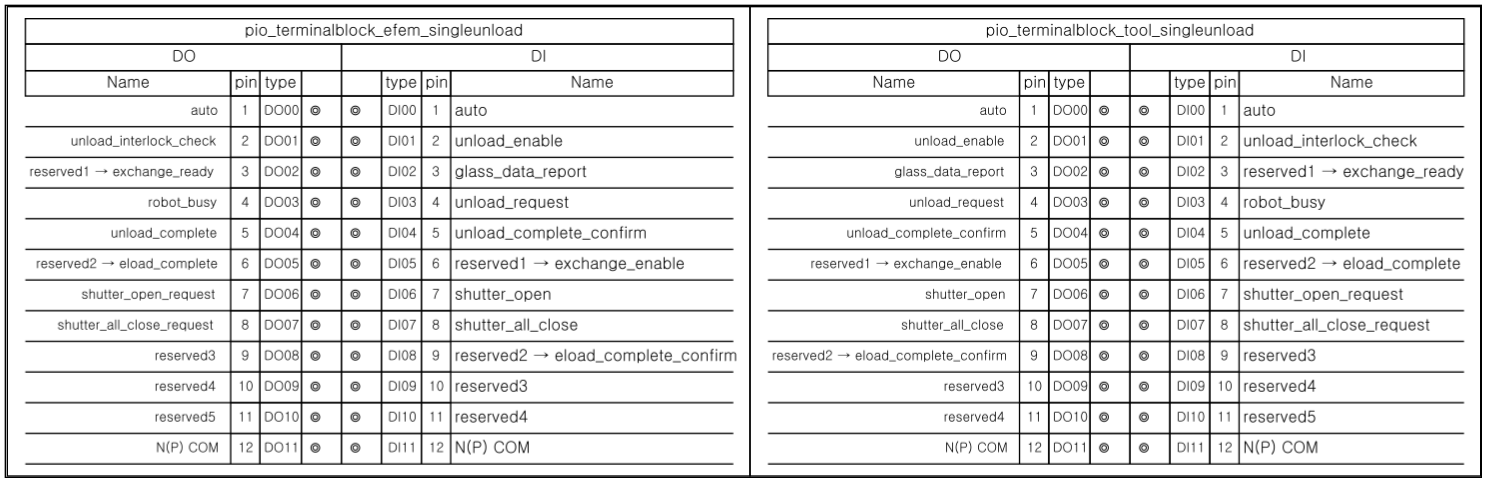


Figure 20 PIO terminal block Single Stage Unload

##### 2.4.6.2.1 PIO Items for Single Stage Unload EFEM

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | EFEM Auto 상태 |
| 2 | unload\_interlock\_check | 반송 가능신호 - Unload 동작 가능한 상태일 경우 ON한다. - Robot 동작중일 경우 OFF한다 |
| 3 | reserved1 → exchange\_ready | EFEM이 unloading 할때 exchange를 할 수 있음을 설비에게 알림. |
| 4 | robot\_busy | 반송 진행신호 - 로봇 동작이 시작되면 ON, 로봇 arm이 홈 위치에 돌아오면 Off |
| 5 | unload\_complete | 반송 완료신호 -. Robot 동작 중 반송완료 시점(흡착 OFF 완료)에서 ON한다. |
| 6 | reserved2 → eload\_complete |  |
| 7 | shutter\_open\_request | Shutter Open 신호 -' 하류 설비에 shutter를 Open 할 경우 사용 한다. -' Robot이 동작중일 경우 상시 ON 한다 |
| 8 | shutter\_all\_close\_request | Shutter Close 신호 -' 하류 설비에 shutter를 Close 할 경우 사용 한다. -' Robot 동작이 완료 할 경우 ON 한다 |
| 9 | reserved3 | Interlock 신호 추가용 |
| 10 | reserved4 | Interlock 신호 추가용 |
| 11 | reserved5 | Interlock 신호 추가용 |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.6.2.2 PIO Items for Single Stage Unload Tool

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | 공정설비 Auto 상태 |
| 2 | unload\_enable | Unload 동작 가능 신호 - Interlock 확인 후 Unload 동작이 가능 할 경우 ON 한다 |
| 3 | glass\_data\_report | Glass Data 송신 신호 - Robot에 Glass Data를 송신 할 경우 ON한다 |
| 4 | unload\_request | 반송 시작신호 -'Interlock 및 Glass Data 확인 후 Robot 반송 시작 신호 |
| 5 | unload\_complete\_confirm | 반송 완료 확인 신호 -. Robot 동작 완료 신호 확인 후 응답 한다 |
| 6 | reserved1 → exchange\_enable | 설비는 EFEM에 exchange 동작을 수용할 수 있음을 알린다. |
| 7 | shutter\_open | Shutter Open 신호 -' Shutter가 Open일 경우 ON 한다 |
| 8 | shutter\_all\_close | Shutter Close 신호 -' Shutter가 Close일 경우 ON 한다 |
| 9 | reserved2 → eload\_complete\_confirm | 설비는 EFEM에 exchange loading한 글래스를 확인했음을 알린다. |
| 10 | reserved3 | Interlock 신호 추가용 |
| 11 | reserved4 | Interlock 신호 추가용 |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.6.2.3 Timing chart for Single Stage Unload without shutter

###### 2.4.6.2.3.1 언로딩 시퀀스중 eloading을 하는 경우

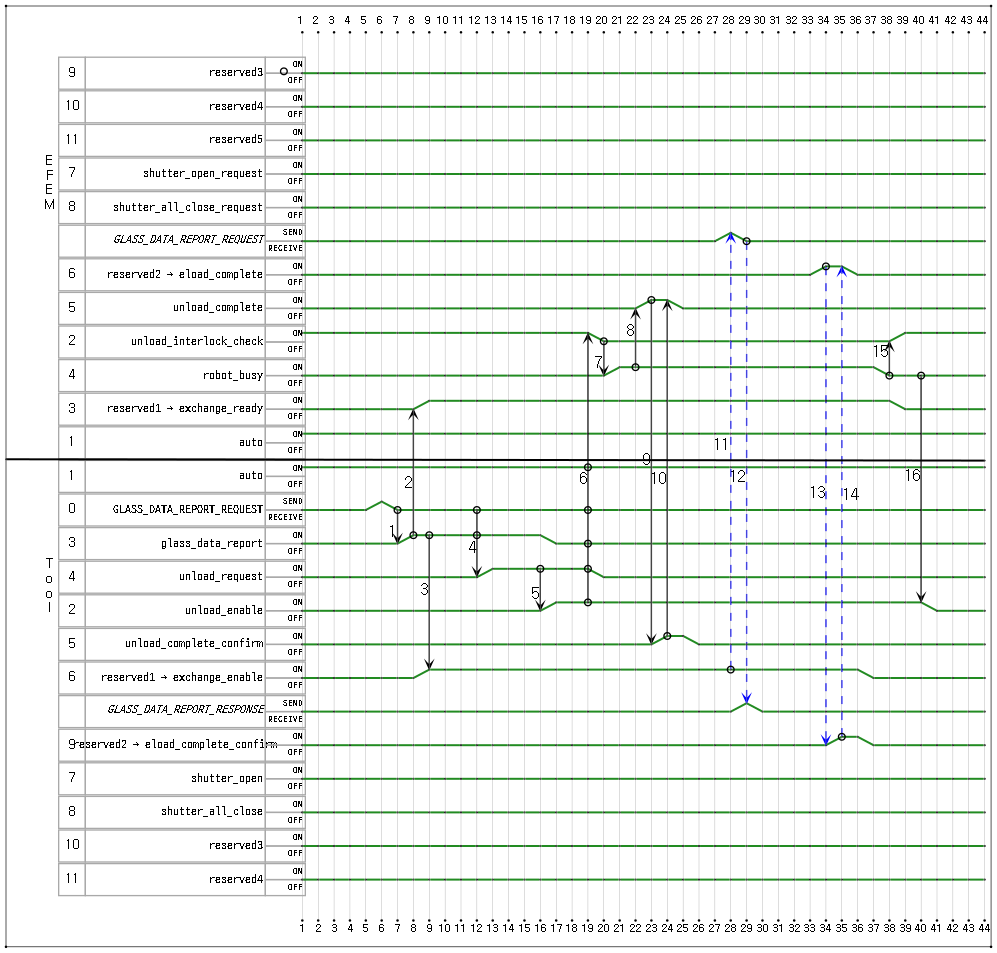


Figure 21 exchange enable bit on 상태에서는 언로딩 중 eloading을 수행함

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off | eq는 efem에게 GLASS\_DATA\_REPORT\_REQUEST 메시지를 보낸 후 glass\_data\_report bit를 on한다 |
| 2 | EFEM reserved1 → exchange\_ready bit On | eq's glass\_data\_report: On | efem은 exchange를 할 수 있는 여분의 글래스를 가지고 갈 수 있으면 exchange\_ready를 on한다 |
| 3 | Tool reserved1 → exchange\_enable bit On | eq's glass\_data\_report: On | Tool 내부에서 exchange loading 글래스를 받지 못할 상황에서는 exchaange\_enable bit를 off 한다. 그렇지 않으면 기본 on으로 유지한다. |
| 4 | Tool unload\_request bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: On |  |
| 5 | Tool unload\_enable bit On | eq's unload\_request: On | eq는 내부의 사정에 따라서 unload\_enable을 on 한다 |
| 6 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_enable: On  eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: Off  eq's unload\_request: On | efem은 eq의 unload\_enable bit on을 확인하고 진입을 위해서 인터락을 해제한다 |
| 7 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 인터락을 해제하고 로봇이 eq영역으로 진입하면서 robot\_busy를 on한다 |
| 8 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 글래스를 픽업해서 영역을 빠져 나오면서 unload\_complete을 on한다 |
| 9 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | eq는 efem의 unload\_comlete bit on을 받고 confirm bit를 켠다 |
| 10 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | eq로 부터 unload\_complete\_confirm bit on을 받으면 efem은 unload\_complete bit를 off한다 |
| 11 | EFEM GLASS\_DATA\_REPORT\_REQUEST message | eq's reserved1 → exchange\_enable: On | efem은 Tool이 exchange 가능하다고 한 상태면 로딩을 위해서 글래스 정보를 Tool에 전달한다 |
| 12 | Tool GLASS\_DATA\_REPORT\_RESPONSE message | efem's GLASS\_DATA\_REPORT\_REQUEST: Send | Tool은 회신한다 |
| 13 | Tool reserved2 → eload\_complete\_confirm bit On | efem's reserved2 → eload\_complete: On | efem은 글래스 로딩을 완료하고 complete bit를 on한다 |
| 14 | EFEM reserved2 → eload\_complete bit Off | eq's reserved2 → eload\_complete\_confirm: On | efem은 Tool의 confirm을 확인하고 complete off한다 |
| 15 | EFEM unload\_interlock\_check bit On | efem's robot\_busy: Off |  |
| 16 | Tool unload\_enable bit Off | efem's robot\_busy: Off  efem's load\_interlock\_check: Off | eq의 unload\_enable을 off한다 |

###### 2.4.6.2.3.2 언로딩 시퀀스중 eloading을 하지 않는 경우

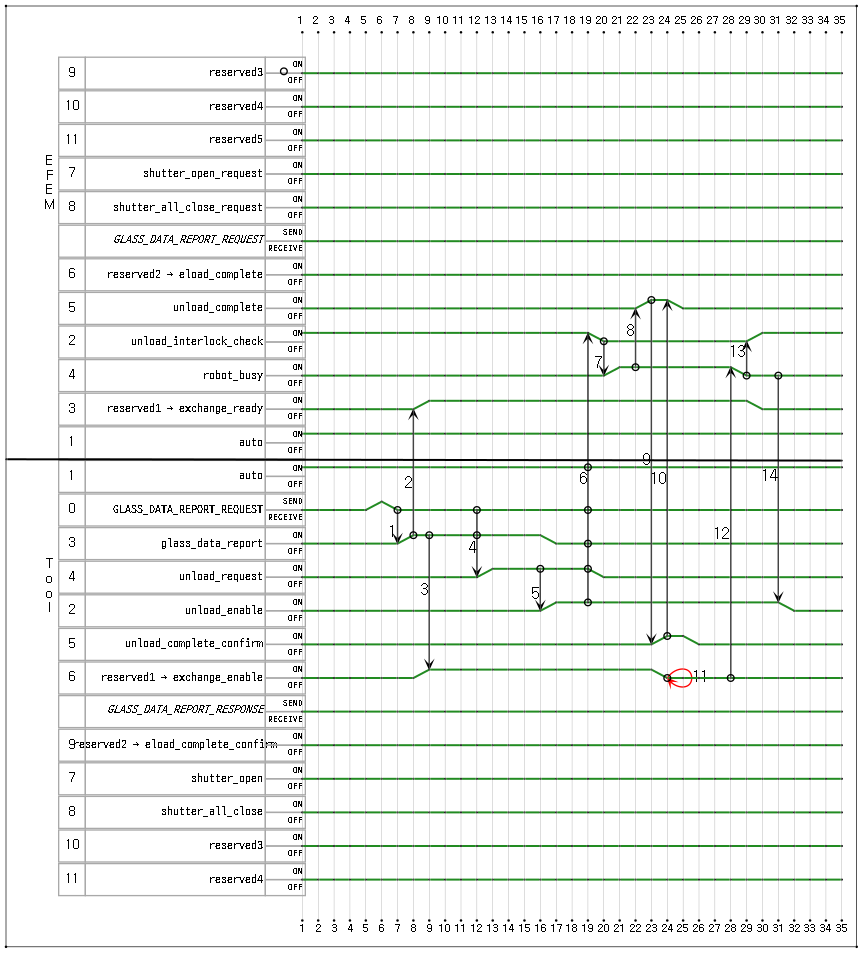


Figure 22 Tool이 exchange\_enable을 off함으로 eloading을 하지 않는 시퀀스

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off | eq는 efem에게 GLASS\_DATA\_REPORT\_REQUEST 메시지를 보낸 후 glass\_data\_report bit를 on한다 |
| 2 | EFEM reserved1 → exchange\_ready bit On | eq's glass\_data\_report: On | efem은 exchange를 할 수 있는 여분의 글래스를 가지고 갈 수 있으면 exchange\_ready를 on한다 |
| 3 | Tool reserved1 → exchange\_enable bit On | eq's glass\_data\_report: On | Tool 내부에서 exchange loading 글래스를 받지 못할 상황에서는 exchaange\_enable bit를 off 한다. 그렇지 않으면 기본 on으로 유지한다. |
| 4 | Tool unload\_request bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: On |  |
| 5 | Tool unload\_enable bit On | eq's unload\_request: On | eq는 내부의 사정에 따라서 unload\_enable을 on 한다 |
| 6 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_enable: On  eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: Off  eq's unload\_request: On | efem은 eq의 unload\_enable bit on을 확인하고 진입을 위해서 인터락을 해제한다 |
| 7 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 인터락을 해제하고 로봇이 eq영역으로 진입하면서 robot\_busy를 on한다 |
| 8 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 글래스를 픽업해서 영역을 빠져 나오면서 unload\_complete을 on한다 |
| 9 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | eq는 efem의 unload\_comlete bit on을 받고 confirm bit를 켠다 |
| 10 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | eq로 부터 unload\_complete\_confirm bit on을 받으면 efem은 unload\_complete bit를 off한다 |
| 11 | Tool reserved1 → exchange\_enable bit Off | eq's reserved1 → exchange\_enable: Off | Tool이 글래스를 받을 수 없는 상황이면 exchange\_eanble off 한다 |
| 12 | EFEM robot\_busy bit Off | eq's reserved1 → exchange\_enable: Off | Tool이 exchange\_enable을 off 하면 efem은 글래스 로딩을 하지 않고 그냥 빠져 나온다 |
| 13 | EFEM unload\_interlock\_check bit On | efem's robot\_busy: Off |  |
| 14 | Tool unload\_enable bit Off | efem's robot\_busy: Off  efem's load\_interlock\_check: Off | eq의 unload\_enable을 off한다 |

##### 2.4.6.2.4 Timing chart for Single Stage Unload with shutter

###### 2.4.6.2.4.1 언로딩 시퀀스중 eloading을 하는 경우

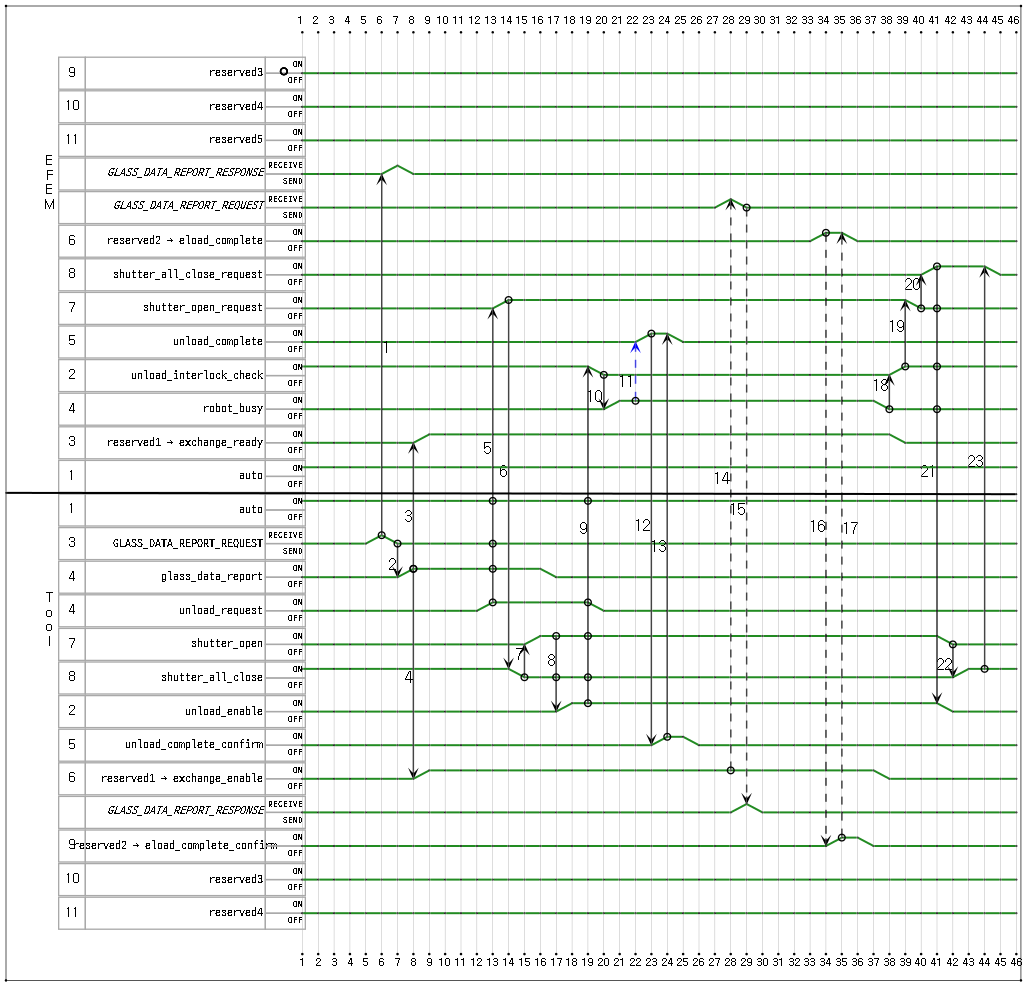


Figure 23 exchange enable bit on 상태에서는 언로딩 중 eloading을 수행함

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | EFEM GLASS\_DATA\_REPORT\_RESPONSE message Send | eq's GLASS\_DATA\_REPORT\_REQUEST: On | Tool은 언로드할 글래스의 정보를 efem에 먼저 보낸다 |
| 2 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off | eq는 언로드할 glass data를 소켓통신으로 efem에 보내고 glass\_data\_bit를 on한다 |
| 3 | EFEM reserved1 → exchange\_ready bit On | eq's glass\_data\_report: On | efem은 exchange를 할 수 있는 여분의 글래스를 가지고 갈 수 있으면 exchange\_ready를 on한다 |
| 4 | Tool reserved1 → exchange\_enable bit On | eq's glass\_data\_report: On | Tool 내부에서 exchange loading 글래스를 받지 못할 상황에서는 exchaange\_enable bit를 off 한다. 그렇지 않으면 기본 on으로 유지한다. |
| 5 | EFEM shutter\_open\_request bit On | eq's auto: On  eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: On  eq's unload\_request: On | efem은 eq의 unload\_request bit가 on 되면 셔터 오픈을 요청한다 |
| 6 | Tool shutter\_all\_close bit Off | efem's shutter\_open\_request: On | eq는 바로 shutter\_all\_close 라는 bit를 하나 더 거쳐서 셔터 오픈을 진행한다 |
| 7 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off | tool은 shutter\_all\_close bit를 off한 후 shutter를 open 한 후 shutter\_open bit를 on한다 |
| 8 | Tool unload\_enable bit On | eq's shutter\_all\_close: Off  eq's shutter\_open: On |  |
| 9 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_request\_stage: Off  eq's unload\_enable\_stage: Off  eq's unload\_enable: On  eq's unload\_request: On  eq's shutter\_open: On  eq's shutter\_all\_close: Off  efem's unload\_request\_stage: Off | efem은 로봇이 handoff conflict 영역을 진입하기 위해서 인터락을 해제한다 |
| 10 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 11 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 글래스를 픽업해서 영역을 빠져 나오면서 unload\_complete을 on한다 |
| 12 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | eq는 efem의 unload\_comlete bit on을 받고 confirm bit를 켠다 |
| 13 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | eq로 부터 unload\_complete\_confirm bit on을 받으면 efem은 unload\_complete bit를 off한다 |
| 14 | EFEM GLASS\_DATA\_REPORT\_REQUEST message | eq's reserved1 → exchange\_enable: On | efem은 Tool이 exchange 가능하다고 한 상태면 로딩을 위해서 글래스 정보를 Tool에 전달한다 |
| 15 | Tool GLASS\_DATA\_REPORT\_RESPONSE message | efem's GLASS\_DATA\_REPORT\_REQUEST: Send | Tool은 회신한다 |
| 16 | Tool reserved2 → eload\_complete\_confirm bit On | efem's reserved2 → eload\_complete: On | efem은 글래스 로딩을 완료하고 complete bit를 on한다 |
| 17 | EFEM reserved2 → eload\_complete bit Off | eq's reserved2 → eload\_complete\_confirm: On | efem은 Tool의 confirm을 확인하고 complete off한다 |
| 18 | EFEM unload\_interlock\_check bit On | efem's robot\_busy: Off |  |
| 19 | EFEM shutter\_open\_request bit Off | efem's unload\_interlock\_check: On | efem은 shutter open request bit를 off한다 |
| 20 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off | efem은 shutter open request bit를 off 한 후 shutter all close reqeust bit를 on 한다 |
| 21 | Tool unload\_enable bit Off | efem's shutter\_all\_close\_request: On  efem's shutter\_open\_request: Off  efem's robot\_busy: Off  efem's unload\_interlock\_check: On  efem's unload\_access\_stage: Off | eq는 unload enable bit를 off 한다 |
| 22 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off |  |
| 23 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On |  |

###### 2.4.6.2.4.2 언로딩 시퀀스중 eloading을 하지 않는 경우

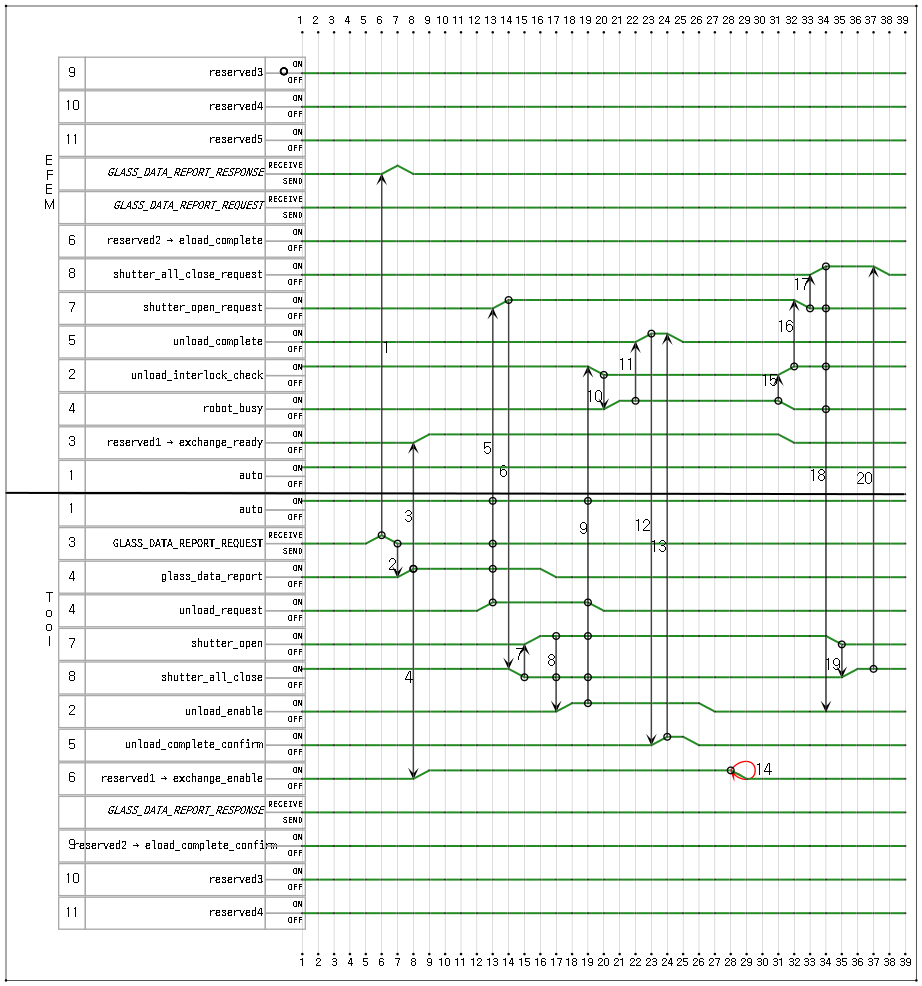


Figure 24 Tool이 exchange\_enable을 off함으로 eloading을 하지 않는 시퀀스

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | EFEM GLASS\_DATA\_REPORT\_RESPONSE message Send | eq's GLASS\_DATA\_REPORT\_REQUEST: On | Tool은 언로드할 글래스의 정보를 efem에 먼저 보낸다 |
| 2 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Off | eq는 언로드할 glass data를 소켓통신으로 efem에 보내고 glass\_data\_bit를 on한다 |
| 3 | EFEM reserved1 → exchange\_ready bit On | eq's glass\_data\_report: On | efem은 exchange를 할 수 있는 여분의 글래스를 가지고 갈 수 있으면 exchange\_ready를 on한다 |
| 4 | Tool reserved1 → exchange\_enable bit On | eq's glass\_data\_report: On | Tool 내부에서 exchange loading 글래스를 받지 못할 상황에서는 exchaange\_enable bit를 off 한다. 그렇지 않으면 기본 on으로 유지한다. |
| 5 | EFEM shutter\_open\_request bit On | eq's auto: On  eq's GLASS\_DATA\_REPORT\_REQUEST: Off  eq's glass\_data\_report: On  eq's unload\_request: On | efem은 eq의 unload\_request bit가 on 되면 셔터 오픈을 요청한다 |
| 6 | Tool shutter\_all\_close bit Off | efem's shutter\_open\_request: On | eq는 바로 shutter\_all\_close 라는 bit를 하나 더 거쳐서 셔터 오픈을 진행한다 |
| 7 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off | tool은 shutter\_all\_close bit를 off한 후 shutter를 open 한 후 shutter\_open bit를 on한다 |
| 8 | Tool unload\_enable bit On | eq's shutter\_all\_close: Off  eq's shutter\_open: On |  |
| 9 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_request\_stage: Off  eq's unload\_enable\_stage: Off  eq's unload\_enable: On  eq's unload\_request: On  eq's shutter\_open: On  eq's shutter\_all\_close: Off  efem's unload\_request\_stage: Off | efem은 로봇이 handoff conflict 영역을 진입하기 위해서 인터락을 해제한다 |
| 10 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 11 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 글래스를 픽업해서 영역을 빠져 나오면서 unload\_complete을 on한다 |
| 12 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | eq는 efem의 unload\_comlete bit on을 받고 confirm bit를 켠다 |
| 13 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | eq로 부터 unload\_complete\_confirm bit on을 받으면 efem은 unload\_complete bit를 off한다 |
| 14 | Tool reserved1 → exchange\_enable bit Off | eq's reserved1 → exchange\_enable: On | tool은 exhange loading을 하면 안되는 상황일때 exchange\_enable bit을 off한다 |
| 15 | EFEM unload\_interlock\_check bit On | efem's robot\_busy: On |  |
| 16 | EFEM shutter\_open\_request bit Off | efem's unload\_interlock\_check: On | efem은 shutter open request bit를 off한다 |
| 17 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off | efem은 shutter open request bit를 off 한 후 shutter all close reqeust bit를 on 한다 |
| 18 | Tool unload\_enable bit Off | efem's shutter\_all\_close\_request: On  efem's shutter\_open\_request: Off  efem's robot\_busy: Off  efem's unload\_interlock\_check: On  efem's unload\_access\_stage: Off | eq는 unload enable bit를 off 한다 |
| 19 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off |  |
| 20 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On |  |

### 2.4.7 PIO for Multi Stage

EFEM이 Tool로 글래스 전달할때 내부 버퍼의 stage 번호를 미리 지정하고 글래스를 투입한다. Shutter는 존재할 수 있다.

#### 2.4.7.1 PIO for Multi Stage Load

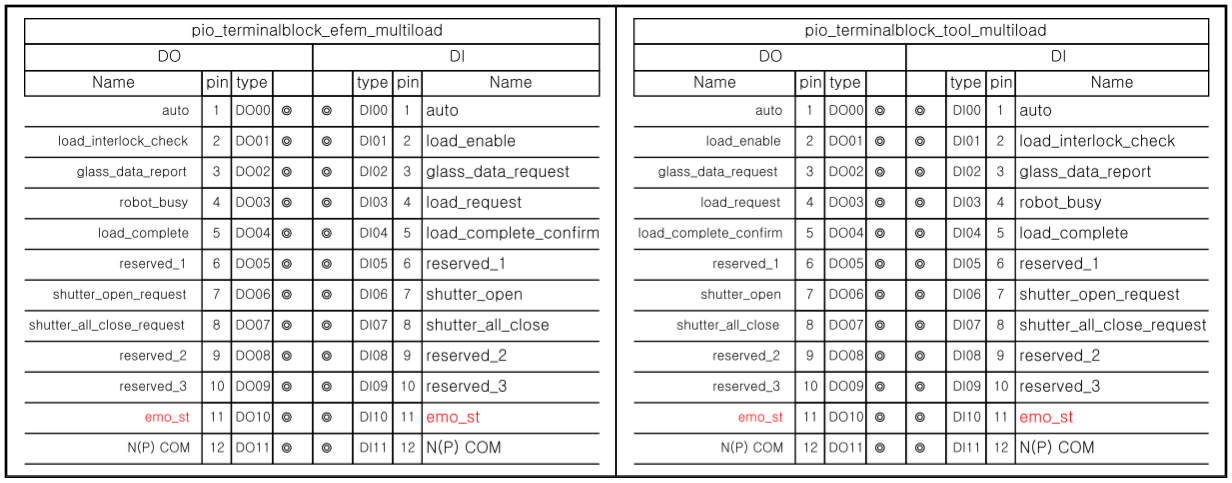


Figure 25 PIO signals for Multi Stage Load

##### 2.4.7.1.1 PIO Items for Multi Stage Load EFEM

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | EFEM Auto 상태  - EFEM Auto 상태로 PIO 준비 완료 시 ON한다.  - 상시 ON |
| 2 | load\_interlock\_check | 반송 가능신호  - Load 동작 가능한 상태일 경우 ON한다.  - Robot 동작중 일 경우 OFF 한다. |
| 3 | glass\_data\_report | Glass Data 송신 신호  - 배출 Glass Data가 확정된 시점에 Glass Data Report 신호를 ON 한다. |
| 4 | robot\_busy | 반송 진행신호  - 로봇 동작이 시작되면 ON, 로봇 위치가 핸드오프 충돌 영역 밖에 있을 때 OFF. |
| 5 | load\_complete | 반송 완료 신호  - 반송완료 시점에 ON 한다. |
| 6 | reserved\_1 |  |
| 7 | shutter\_open\_request | Shutter Open  - Robot 동작 전 설비의 Shutter를 Open 하기 위해 신호를 ON 한다. |
| 8 | shutter\_all\_close\_request | Shutter Close  - Robot 동작 완료 후 설비의 Shutter를 Close 하기 위해서 신호를 ON 한다. |
| 9 | reserved\_2 | Interlock 신호 추가용For adding interlock signal |
| 10 | reserved\_3 | Interlock 신호 추가용For adding interlock signal |
| 11 | emo\_st | EFEM Emergency |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.7.1.2 PIO Items for Multi Stage Load Tool

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | 공정설비 Auto 상태  - 공정설비 Auto 상태로 PIO 준비 완료 시 ON한다.  - 상시 ON |
| 2 | load\_enable | 투입가능 신호  - AUTO MODE 에서 수신 반송 가능 시 ON 한다. |
| 3 | glass\_data\_request | Glass Data 요구신호  - Glass를 받기 전에 Data를 요구하여 확인한다. |
| 4 | load\_request | 투입요구 신호  - Robot으로 부터 투입 요구를 한다. |
| 5 | load\_complete\_confirm | 반송 완료 확인 신호  -. Robot Complete 신호가 ON 되면 인터락 확인 후 신호를 ON 한다. |
| 6 | reserved\_1 |  |
| 7 | shutter\_open | Shutter Open 신호  -' Shutter가 Open일 경우 ON 한다 |
| 8 | shutter\_all\_close | Shutter Close 신호  -' Shutter가 Close일 경우 ON 한다 |
| 9 | reserved\_2 | Interlock 신호 추가용 |
| 10 | reserved\_3 | Interlock 신호 추가용 |
| 11 | emo\_st | EQ Emergency |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.7.1.3 Timing chart for Multi Stage Load

...

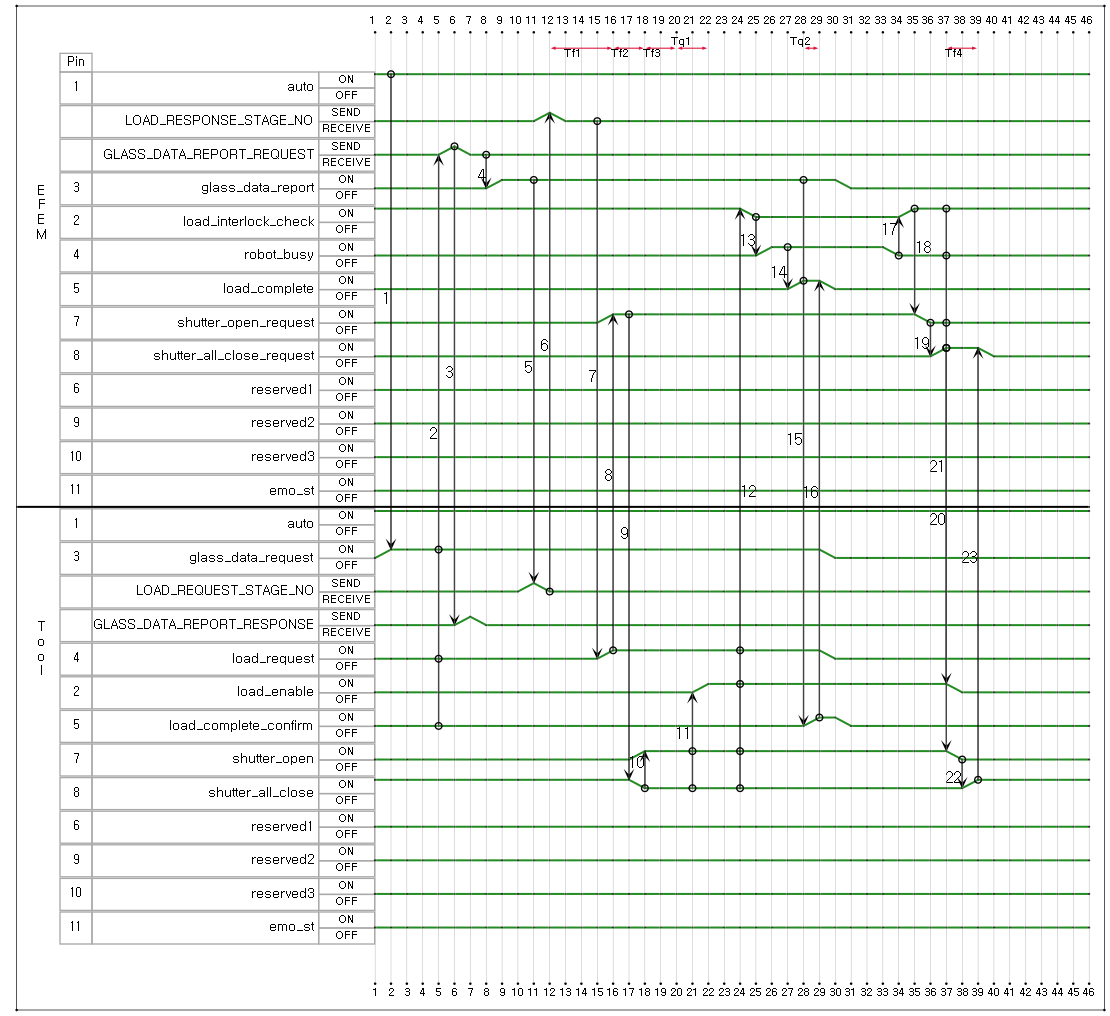


Figure 26 멀티 Stage의 Load 타임 차트

###### 2.4.7.1.3.1 차트 시퀀스 설명

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool glass\_data\_request bit On | efem's auto: On | 로딩 시퀀스의 시작은 eq의 glass\_data\_request bit on으로 시작한다. |
| 2 | EFEM GLASS\_DATA\_REPORT\_REQUEST message Send | eq's load\_complete\_confirm: Off  eq's load\_request: Off  eq's glass\_data\_request: On | efem은 tool의 bit 상태를 확인하고 GLASS\_DATA\_REPORT\_REQUEST 소켓 메시지를 전송한다 |
| 3 | EFEM glass\_data\_report bit On | efem's GLASS\_DATA\_REPORT\_REQUEST: Send | efem의 glass\_data\_report bit는 GLASS\_DATA\_REPORT\_REQUEST 소켓 메시지를 보낸 직후 켜고 load\_complete\_bit가 완전히 꺼졌을때 끈다 |
| 4 | Tool LOAD\_REQUEST\_STAGE\_NO message Send | efem's glass\_data\_report: On | eq는 efem이 전송한 글래스 데이터를 확인하고 LOAD\_REQUEST\_STAGE\_NO 메시지를 efem에게 전송한다 |
| 5 | EFEM LOAD\_RESPONSE\_STAGE\_NO message Send | eq's LOAD\_REQUEST\_STAGE\_NO: Send | eq가 특정 slot no에 글래스 로드할 것을 요청하면 efem은 특정 stage no에서 로드할 것이라고 응답하며 해당 stage 번호는 유효한 값으로 유지한다 |
| 6 | Tool load\_request bit On | efem's LOAD\_RESPONSE\_STAGE\_NO: Send | eq와 efem이 서로 목표 slot 번호를 교환하면 eq는 load\_request bit을 on한다 |
| 7 | EFEM shutter\_open\_request bit On | eq's load\_request: On | efem은 eq의 load\_reqeust를 확인한 후 셔터를 열기 위한 요청을 시작한다 |
| 8 | Tool shutter\_all\_close bit Off | efem's shutter\_open\_request: On | 셔터를 열때 eq는 efem의 shutter\_open\_request bit를 직접적으로 보고 움직이는 것이 아니라 내부의 shutter\_all\_close bit의 꺼짐을 보고 shutter open 동작을 한 후 shutter\_open bit를 켜게 된다 |
| 9 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off |  |
| 10 | Tool load\_enable bit On | eq's shutter\_all\_close: Off  eq's shutter\_open: On | 기존에는 LOAD\_ENALBLE\_STAGE 소켓 메시지의 수신을 체크했으나, 복잡해서 삭제하고, 셔터 상태만 확인하고 eq는 load\_enable 한다. |
| 11 | EFEM load\_interlock\_check bit Off | eq's shutter\_all\_close: Off  eq's shutter\_open: On  eq's load\_request: On  eq's load\_enable: On | interlock은 안전을 위한 추가적인 절차이다 |
| 12 | EFEM robot\_busy bit On | efem's load\_interlock\_check: Off | 글래스를 든 로봇이 eq로 진입하면서 busy bit를 on한다 |
| 13 | EFEM load\_complete bit On | efem's robot\_busy: On |  |
| 14 | Tool load\_complete\_confirm bit On | efem's glass\_data\_report: On  efem's load\_complete: On | complete confirm은 서로 글래스의 교환을 확인하는 과정이며 글래스 감지 기능이 없는 eq의 경우는 반드시 확인되어야 한다 |
| 15 | EFEM load\_complete bit Off | eq's load\_complete\_confirm: On | efem은 load\_complete\_comfirm이 on되는 것을 확인하고 load\_complete를 off한다 |
| 16 | EFEM load\_interlock\_check bit On | efem's robot\_busy: Off |  |
| 17 | EFEM shutter\_open\_request bit Off | efem's load\_interlock\_check: On |  |
| 18 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off | efem은 eq에 셔터를 닫을 것을 요청한다 |
| 19 | Tool load\_enable bit Off | efem's shutter\_all\_close\_request: On | eq는 efem의 shutter\_all\_close\_request가 on되는 것을 확인하면 load\_eanble을 off 한다 |
| 20 | Tool shutter\_open bit Off | efem's load\_interlock\_check: On  efem's robot\_busy: Off  efem's shutter\_open\_request: Off  efem's shutter\_all\_close\_request: On | eq는 efem의 요청에 의해서 shutter\_open bit을 off하여 셔터를 닫는다 |
| 21 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off | eq의 셔터가 실제로 닫혔는지를 eq:shutter\_all\_close bit off로 확인한다 |
| 22 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On |  |

#### 2.4.7.2 PIO for Multi Stage Unload

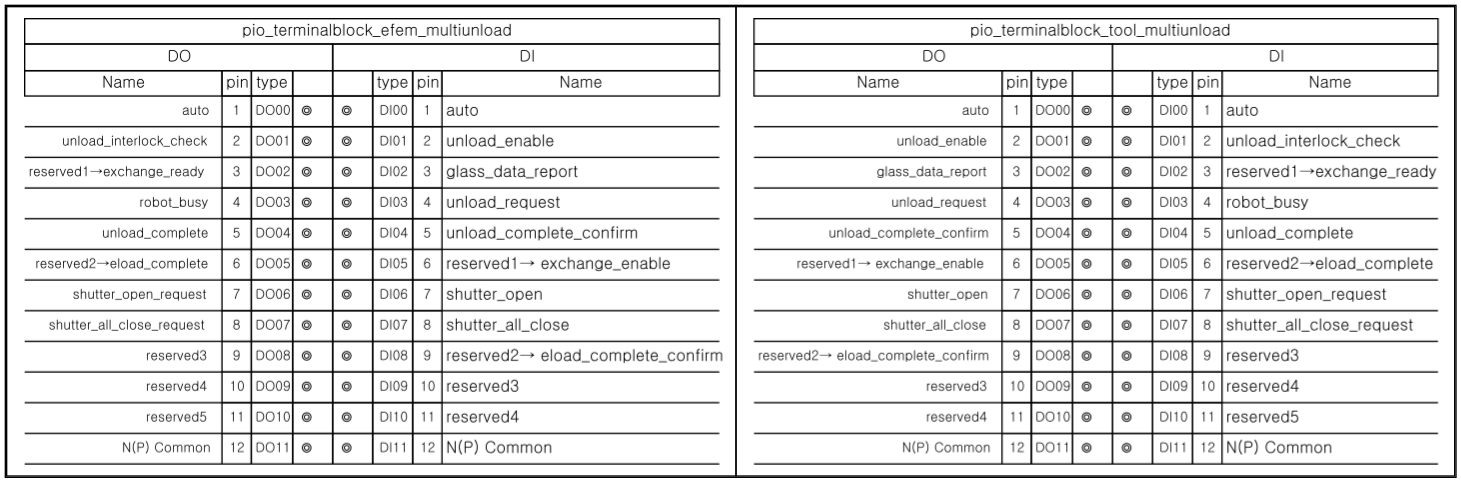


Figure 27 PIO signals for Multi Stage UnLoad

##### 2.4.7.2.1 PIO Items for Multi Stage Unload EFEM

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | EFEM Auto 상태  - EFEM Auto 상태로 PIO 준비 완료 시 ON한다.  - 상시 ON |
| 2 | unload\_interlock\_check | 반송 가능신호  -. 인터락 확인 후 ON 시킨다. |
| 3 | reserved1→exchange\_ready | 최초에는 unload\_demand 였는데 사용안하면서 reserved로 변경하였고, 이후 exchange\_ready로 변경함. |
| 4 | robot\_busy | 반송 진행신호. 로봇 동작이 시작되면 ON, 로봇 위치가 핸드오프 충돌 영역 밖에 있을 때 OFF. |
| 5 | unload\_complete | 반송 완료신호. 반송완료 시점에 ON한다. |
| 6 | reserved2→eload\_complete |  |
| 7 | shutter\_open\_request | Shutter Open  -' 로봇 동작 전 설비의 셔터를 Open 하기 위해 신호를 ON 한다. |
| 8 | shutter\_all\_close\_request | Shutter Close  -' Robot 동작 완료 후 설비의 Shutter를 Close 하기 위해 신호를 ON 한다. |
| 9 | reserved3 | Interlock 신호 추가용 |
| 10 | reserved4 | Interlock 신호 추가용 |
| 11 | reserved5 | Interlock 신호 추가용 |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.7.2.2 PIO Items for Multi Stage Unload Tool

| No | Signal Name | Usage |
| --- | --- | --- |
| 1 | auto | 공정설비 Auito 상태  - 공정설비 Auto 상태로 PIO 준비 완료 시 ON한다.  - 상시 ON |
| 2 | unload\_enable | 배출 가능 신호  - 인터락 확인 후 배출이 가능 할 때 ON 시킨다. |
| 3 | glass\_data\_report | 배출 Glass Data 송신 신호  - 배출 가능한 Glass의 Data를 Robot측으로 송신한다. |
| 4 | unload\_request | 배출 요구 신호  - 다단 장치의 배출이 가능 할 경우 ON 시킨다. |
| 5 | unload\_complete\_confirm | 반송 완료 확인 신호  - Robot Complete 신호가 ON되면 인터락 확인 후 신호를 ON 시킨다. |
| 6 | reserved1→ exchange\_enable |  |
| 7 | shutter\_open | Shutter Open신호  -. Shutter가 Open 되면 신호를 ON한다. |
| 8 | shutter\_all\_close | Shutter Close신호  -. Shutter가 Close 되면 신호를 ON한다. |
| 9 | reserved2→ eload\_complete\_confirm | Interlock 신호 추가용 |
| 10 | reserved3 | Interlock 신호 추가용 |
| 11 | reserved4 | Interlock 신호 추가용 |
| 12 | N(P) Common | Relay 배선을 위한 N(P) Com 신호 결선 |

##### 2.4.7.2.3 Timing chart for Multi Stage Unload

###### 2.4.7.2.3.1 언로딩 시퀀스중 eloading을 하는 경우

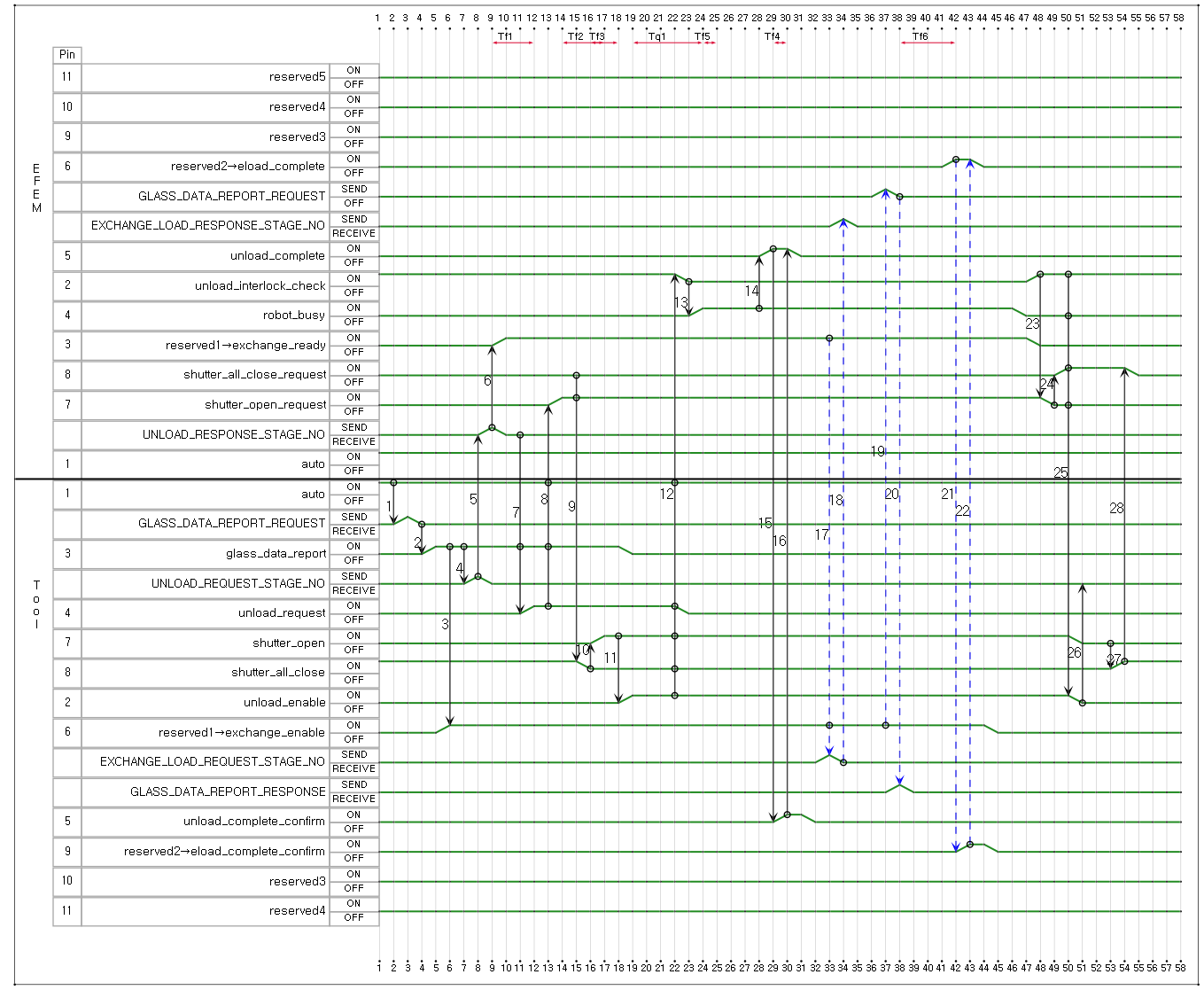


Figure 28 멀티 Stage의 Unload 타임 차트

2.4.7.2.3.1.1 타임차트 시퀀스 설명

| No. | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool GLASS\_DATA\_REPORT\_REQUEST message Send | eq's auto: On | eq는 언로드할 glass data를 소켓통신으로 efem에 보낸다 |
| 2 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Send |  |
| 3 | Tool reserved1→exchange\_enable bit On | eq's glass\_data\_report: On | Tool 내부에서 exchange loading 글래스를 받지 못할 상황에서는 exchaange\_enable bit를 off 한다. 그렇지 않으면 기본 on으로 유지한다. |
| 4 | Tool UNLOAD\_REQUEST\_STAGE\_NO message Send | eq's glass\_data\_report: On | eq가 UNLOAD\_REQUEST\_STAGE\_NO를 efem에 보내서 특정 slot no에서 언로드할 것을 요청한다 |
| 5 | EFEM UNLOAD\_RESPONSE\_STAGE\_NO message Send | eq's UNLOAD\_REQUEST\_STAGE\_NO: Send | eq가 특정 slot no에서 언로드할 것을 요청하면 efem은 특정 stage no에서 언로드할 것이라고 응답하며 해당 stage 번호는 유효한 값으로 유지한다 |
| 6 | EFEM reserved1→exchange\_ready bit On | efem's UNLOAD\_RESPONSE\_STAGE\_NO: Send | efem은 exchange를 할 수 있는 여분의 글래스를 가지고 갈 수 있으면 exchange\_ready를 on한다 |
| 7 | Tool unload\_request bit On | efem's UNLOAD\_RESPONSE\_STAGE\_NO: Send  eq's glass\_data\_report: On | eq는 efem으로부터 UNLOAD\_RESPONSE\_STAGE\_NO 메시지가 오면 unload\_request bit를 on한다 |
| 8 | EFEM shutter\_open\_request bit On | eq's auto: On  eq's glass\_data\_report: On  eq's unload\_request: On | efem은 eq의 unload\_request bit가 on 되면 셔터 오픈을 요청한다 |
| 9 | Tool shutter\_all\_close bit Off | efem's shutter\_all\_close\_request: Off  efem's shutter\_open\_request: On |  |
| 10 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off | tool은 shutter\_all\_close bit를 off한 후 shutter를 open 한 후 shutter\_open bit를 on한다 |
| 11 | Tool unload\_enable bit On | eq's shutter\_open: On | tool은 unload\_enable bit를 on한다 |
| 12 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_enable: On  eq's unload\_request: On  eq's shutter\_open: On  eq's shutter\_all\_close: Off | efem은 로봇이 handoff conflict 영역을 진입하기 위해서 인터락을 해제한다 |
| 13 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 14 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 15 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | efem의 unload\_complete가 on되면 tool은 complete confirm을 한다 |
| 16 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | efem은 tool의 unload complete confirm 이 on 되기 시작하면 unload complete를 off한다 |
| 17 | Tool EXCHANGE\_LOAD\_REQUEST\_STAGE\_NO message | efem's reserved1→exchange\_ready: On  eq's reserved1→exchange\_enable: On | eq는 로딩할 stage 번호를 efem에 전달한다 |
| 18 | EFEM EXCHANGE\_LOAD\_RESPONSE\_STAGE\_NO message | eq's EXCHANGE\_LOAD\_REQUEST\_STAGE\_NO: Send |  |
| 19 | EFEM GLASS\_DATA\_REPORT\_REQUEST message | eq's reserved1→exchange\_enable: On | efem은 exchange 가능하면 글래스 로딩을 위해서 GLASS\_DATA\_REPORT\_REQUEST를 Tool로 보낸다 |
| 20 | Tool GLASS\_DATA\_REPORT\_RESPONSE message | efem's GLASS\_DATA\_REPORT\_REQUEST: Send |  |
| 21 | Tool reserved2→eload\_complete\_confirm bit On | efem's reserved2→eload\_complete: On | efem이 미리 들고 온 글래스를 로딩완료하고 load\_complete를 on하면 eq는 load complete confirm bit on한다 |
| 22 | EFEM reserved2→eload\_complete bit Off | eq's reserved2→eload\_complete\_confirm: On | eq의 load\_complete\_confirm bit on 확인 후 efem은 complete bit off 한다 |
| 23 | EFEM shutter\_open\_request bit Off | efem's unload\_interlock\_check: On | efem은 shutter open request bit를 off한다 |
| 24 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off | efem은 shutter open request bit를 off 한 후 shutter all close reqeust bit를 on 한다 |
| 25 | Tool unload\_enable bit Off | efem's robot\_busy: Off  efem's unload\_interlock\_check: On  efem's shutter\_all\_close\_request: On  efem's shutter\_open\_request: Off | eq는 efem의 shutter\_all\_close\_request bit의 on을 받고, shutter\_open도 off하고, unload enable bit를 off 한다 |
| 26 | Tool UNLOAD\_REQUEST\_STAGE\_NO message | eq's unload\_enable: Off |  |
| 27 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off | eq는 shutter\_all\_close를 on한다 |
| 28 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On | efem은 shutter\_all\_close\_request를 off한다 |

###### 2.4.7.2.3.2 언로딩 시퀀스중 eloading을 하지 않는 경우

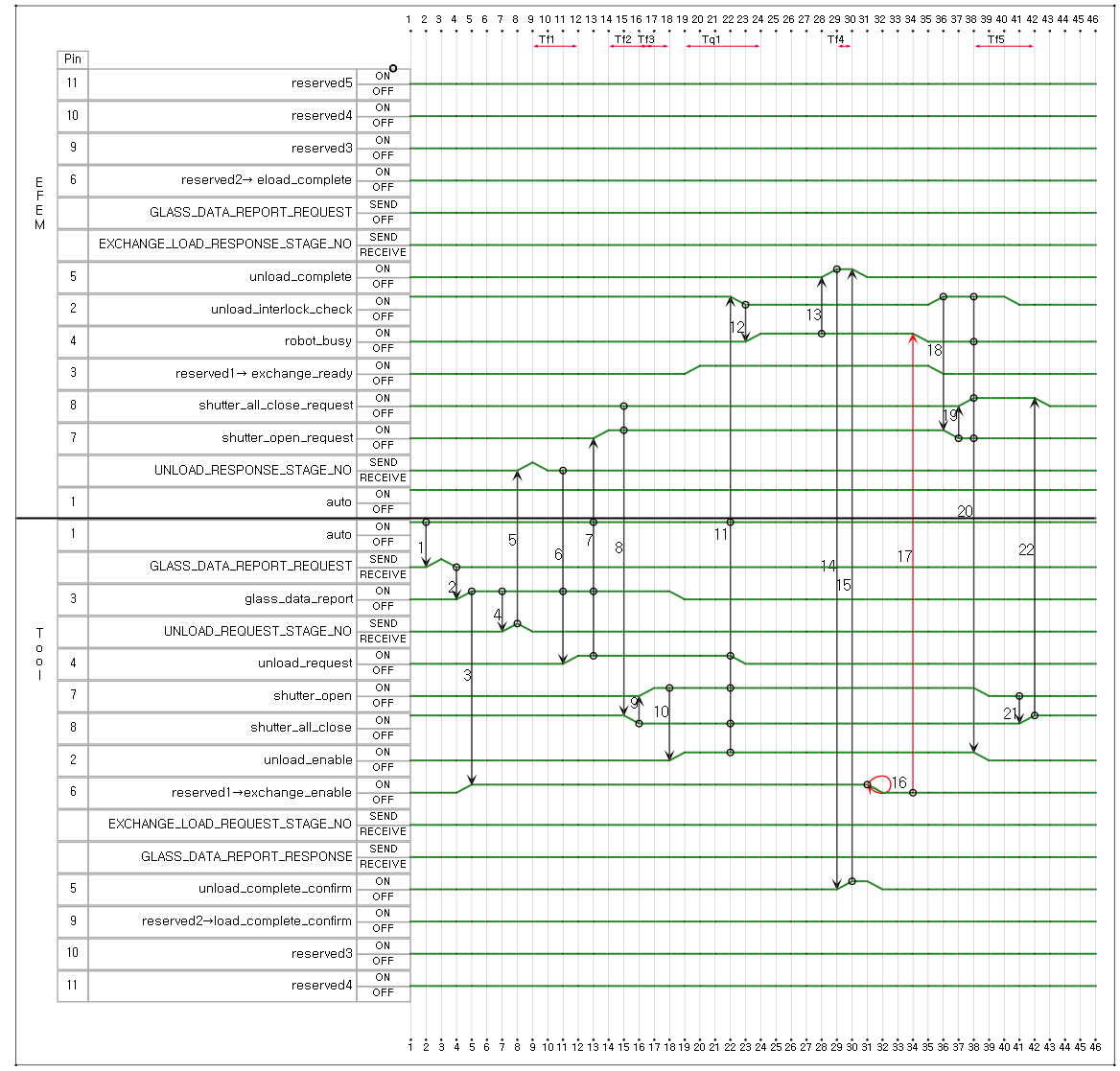


Figure 29 Tool이 exchange\_enable을 off함으로 eloading을 하지 않는 시퀀스

2.4.7.2.3.2.1 타임차트 시퀀스 설명

| No | Action | PreCondition | 설명 |
| --- | --- | --- | --- |
| 1 | Tool GLASS\_DATA\_REPORT\_REQUEST message Send | eq's auto: On | eq는 언로드할 glass data를 소켓통신으로 efem에 보낸다 |
| 2 | Tool glass\_data\_report bit On | eq's GLASS\_DATA\_REPORT\_REQUEST: Send |  |
| 3 | Tool reserved1→exchange\_enable bit On | eq's glass\_data\_report: On | Tool은 언로딩을 시작할때 미리 exhange 가능 여부를 bit on한다 |
| 4 | Tool UNLOAD\_REQUEST\_STAGE\_NO message Send | eq's glass\_data\_report: On | eq가 UNLOAD\_REQUEST\_STAGE\_NO를 efem에 보내서 특정 slot no에서 언로드할 것을 요청한다 |
| 5 | EFEM UNLOAD\_RESPONSE\_STAGE\_NO message Send | eq's UNLOAD\_REQUEST\_STAGE\_NO: Send | eq가 특정 slot no에서 언로드할 것을 요청하면 efem은 특정 stage no에서 언로드할 것이라고 응답하며 해당 stage 번호는 유효한 값으로 유지한다 |
| 6 | Tool unload\_request bit On | efem's UNLOAD\_RESPONSE\_STAGE\_NO: Send  eq's glass\_data\_report: On | eq는 efem으로부터 UNLOAD\_RESPONSE\_STAGE\_NO 메시지가 오면 unload\_request bit를 on한다 |
| 7 | EFEM shutter\_open\_request bit On | eq's auto: On  eq's glass\_data\_report: On  eq's unload\_request: On | efem은 eq의 unload\_request bit가 on 되면 셔터 오픈을 요청한다 |
| 8 | Tool shutter\_all\_close bit Off | efem's shutter\_all\_close\_request: Off  efem's shutter\_open\_request: On |  |
| 9 | Tool shutter\_open bit On | eq's shutter\_all\_close: Off | tool은 shutter\_all\_close bit를 off한 후 shutter를 open 한 후 shutter\_open bit를 on한다 |
| 10 | Tool unload\_enable bit On | eq's shutter\_open: On | tool은 unload\_enable bit를 on한다 |
| 11 | EFEM unload\_interlock\_check bit Off | eq's auto: On  eq's unload\_request\_stage: Off  eq's unload\_enable\_stage: Off  eq's unload\_enable: On  eq's unload\_request: On  eq's shutter\_open: On  eq's shutter\_all\_close: Off  efem's unload\_request\_stage: Off | efem은 로봇이 handoff conflict 영역을 진입하기 위해서 인터락을 해제한다 |
| 12 | EFEM robot\_busy bit On | efem's unload\_interlock\_check: Off | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 13 | EFEM unload\_complete bit On | efem's robot\_busy: On | efem은 로봇을 움직이면서 robot\_busy를 on한다 |
| 14 | Tool unload\_complete\_confirm bit On | efem's unload\_complete: On | efem의 unload\_complete가 on되면 tool은 complete confirm을 한다 |
| 15 | EFEM unload\_complete bit Off | eq's unload\_complete\_confirm: On | efem은 tool의 unload complete confirm 이 on 되기 시작하면 unload complete를 off한다 |
| 16 | Tool reserved1→exchange\_enable bit Off | eq's reserved1→exchange\_enable: On | tool은 exhange loading을 하면 안되는 상황일때 exchange\_enable bit을 off한다 |
| 17 | EFEM robot\_busy bit Off | eq's reserved1→exchange\_enable: Off | efem은 eq의 exchange\_enable bit가 off되면 로봇은 로딩 동작을 진행하지 않고 언로딩만 한 상태로 빠져 나온다 |
| 18 | EFEM shutter\_open\_request bit Off | efem's unload\_interlock\_check: On | efem은 shutter open request bit를 off한다 |
| 19 | EFEM shutter\_all\_close\_request bit On | efem's shutter\_open\_request: Off | efem은 shutter open request bit를 off 한 후 shutter all close reqeust bit를 on 한다 |
| 20 | Tool unload\_enable bit Off | efem's robot\_busy: Off  efem's unload\_interlock\_check: On  efem's shutter\_all\_close\_request: On  efem's shutter\_open\_request: Off | eq는 efem의 shutter\_all\_close\_request bit의 on을 받고, shutter\_open도 off하고, unload enable bit를 off 한다 |
| 21 | Tool shutter\_all\_close bit On | eq's shutter\_open: Off | eq는 shutter\_all\_close를 on한다 |
| 22 | EFEM shutter\_all\_close\_request bit Off | eq's shutter\_all\_close: On | efem은 shutter\_all\_close\_request를 off한다 |

## 2.5 EFEM ↔ Tool local socket

### 2.5.1 개요

#### 2.5.1.1 소켓 구성

1. 초기 사양에는 EFEM, Tool 각각, 송/수신 용도로 Client/Server 소켓을 별도로 가져야 한다고 했지만, 하나의 소켓으로 송/수신이 충분히 가능하다고 판단되고, 고객의 양해를 구하여 EFEM, Tool 각각 하나의 소켓을 생성해서 송신/수신 용으로 사용하도록 한다.

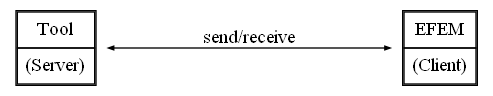


Figure 30 소켓통신 연결 구성도기본 Tool외에 추가로 EFEM이 연결해야 할 설비가 있는 경우 port 번호를 1씩 증가하면서 할당한다.

|  | Server(Tool) | Client(Tool) |
| --- | --- | --- |
| IP Address | 100.100.100.100 |  |
| Port Number | basic 4000  extra 4001, 4002, ... |  |
| Subnet Mask | 0xFFFFFF00 |  |

1. TCP/IP 를 사용하며 EFEM과 Tool간은 지속적으로 통신 연결 상태를 유지해야 한다.

2. 설비 전원 ON/OFF 등 어떤 이유에서 통신연결이 끊어졌을 경우 서버는 가능한 빨리 소켓을 열어야 한다.

3. 클라이언트는 서버와의 통신이 끊어졌을 경우 지속적으로 연결 재시도를 해야 한다.

4. 통신 연결이 되지 않았을 경우 UI를 통해서 작업자에게 알려야 한다.

#### 2.5.1.2 데이터 포맷

1. 주고 받는 데이터의 형식은 모든 정보를 json객체 하나에 넣는 형태로 한다. 기존에는 command가 json 객체의 바깥쪽에 있었고 json 객체에는 command를 제외한 데이터를 포함했지만 데이터 형식의 일관성을 위해서 command도 json 내부에 포함시킨다.

2. 패킷 구분의 안정성을 위해서 데이터의 전/후에 stx/etx를 둔다.

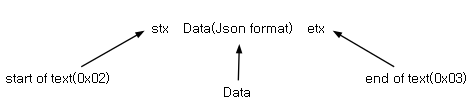


Figure 31 socket-format

| . | . |
| --- | --- |
| sample message | Socket byte stream sample  STX{  "COMMAND": "WORKORDER\_START\_REQEUST",  "WORKSEQUENCEPP": [  {  "WORKSEQPP\_NO": 1,  "WORKSEQ\_NO": 1,  "PPID": "Ppid\_001"  },  {  "WORKSEQPP\_NO": 2,  "WORKSEQ\_NO": 2,  "PPID": "Ppid\_001"  },  {  "WORKSEQPP\_NO": 3,  "WORKSEQ\_NO": 2,  "PPID": "Ppid\_002"  }  ]  }ETX  STX  {  "COMMAND": "MANUAL\_START\_REQUEST"  }ETX  //  0x02, 0x7B, 0x0D, 0x0A, 0x20, 0x20, 0x22, 0x43, 0x4F, 0x4D, 0x4D, 0x41,  0x4E, 0x44, 0x22, 0x3A, 0x20, 0x22, 0x57, 0x4F, 0x52, 0x4B, 0x4F, 0x52,  0x44, 0x45, 0x52, 0x5F, 0x53, 0x54, 0x41, 0x52, 0x54, 0x5F, 0x52, 0x45,  0x51, 0x45, 0x55, 0x53, 0x54, 0x22, 0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x22,  0x57, 0x4F, 0x52, 0x4B, 0x53, 0x45, 0x51, 0x55, 0x45, 0x4E, 0x43, 0x45,  0x50, 0x50, 0x22, 0x3A, 0x20, 0x5B, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20,  0x7B, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x57, 0x4F,  0x52, 0x4B, 0x53, 0x45, 0x51, 0x50, 0x50, 0x5F, 0x4E, 0x4F, 0x22, 0x3A,  0x20, 0x31, 0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22,  0x57, 0x4F, 0x52, 0x4B, 0x53, 0x45, 0x51, 0x5F, 0x4E, 0x4F, 0x22, 0x3A,  0x20, 0x31, 0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22,  0x50, 0x50, 0x49, 0x44, 0x22, 0x3A, 0x20, 0x22, 0x50, 0x70, 0x69, 0x64,  0x5F, 0x30, 0x30, 0x31, 0x22, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x7D,  0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x7B, 0x0D, 0x0A, 0x20, 0x20,  0x20, 0x20, 0x20, 0x20, 0x22, 0x57, 0x4F, 0x52, 0x4B, 0x53, 0x45, 0x51,  0x50, 0x50, 0x5F, 0x4E, 0x4F, 0x22, 0x3A, 0x20, 0x32, 0x2C, 0x0D, 0x0A,  0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x57, 0x4F, 0x52, 0x4B, 0x53,  0x45, 0x51, 0x5F, 0x4E, 0x4F, 0x22, 0x3A, 0x20, 0x32, 0x2C, 0x0D, 0x0A,  0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x50, 0x50, 0x49, 0x44, 0x22,  0x3A, 0x20, 0x22, 0x50, 0x70, 0x69, 0x64, 0x5F, 0x30, 0x30, 0x31, 0x22,  0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x7D, 0x2C, 0x0D, 0x0A, 0x09, 0x7B,  0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x57, 0x4F, 0x52,  0x4B, 0x53, 0x45, 0x51, 0x50, 0x50, 0x5F, 0x4E, 0x4F, 0x22, 0x3A, 0x20,  0x33, 0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x57,  0x4F, 0x52, 0x4B, 0x53, 0x45, 0x51, 0x5F, 0x4E, 0x4F, 0x22, 0x3A, 0x20,  0x32, 0x2C, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x22, 0x50,  0x50, 0x49, 0x44, 0x22, 0x3A, 0x20, 0x22, 0x50, 0x70, 0x69, 0x64, 0x5F,  0x30, 0x30, 0x32, 0x22, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x7D, 0x0D,  0x0A, 0x20, 0x20, 0x5D, 0x0D, 0x0A, 0x7D,0x03,  0x0D, 0x0A,  0x02, 0x7B,  0x0D, 0x0A, 0x20, 0x20, 0x22, 0x43, 0x4F, 0x4D, 0x4D, 0x41, 0x4E, 0x44,  0x22, 0x3A, 0x20, 0x22, 0x4D, 0x41, 0x4E, 0x55, 0x41, 0x4C, 0x5F, 0x53,  0x54, 0x41, 0x52, 0x54, 0x5F, 0x52, 0x45, 0x51, 0x55, 0x45, 0x53, 0x54,  0x22, 0x0D, 0x0A, 0x7D, 0x03,  0x0D, 0x0A |

### 2.5.2 메시지 정의

#### 2.5.2.1 메시지 설명 방법

json schema 참조

1. 기본 type일 경우: string, number, integer 와 같은 기본 타입인 경우는 type으로 끝난다.

2. Object 타입인 경우: properties 를 나열해서 내부에 어떤 property가 있는지 설명하고 각 property의 type을 지정한다.required 를 지정해서 필수 property가 무엇인지를 지정한다.

3. Array 타입인 경우: items 를 나열하여 내부에 어떤 type의 item들이 있는지 설명한다.

4. $ref: 다른 곳에서 정의된 스키마를 참조하도록 지시한다.

Warning

각 메시지 정의에 있는 optional 항목도 내용이 없으면 비워두되 항목은 있어야 한다.

이것은 메시지 해석에 오류가 업게하기 위해서 이다.

#### 2.5.2.2 소켓 메시지들

##### 2.5.2.2.1 EFEM의 작업완료 처리관련 중요 속성

EFEM이 작업완료 처리를 EAP에 보고 하기 위해서 다음의 두 메시지에는 작업완료를 판단할 수 있는 정보가 포함되어야 한다

| message | case | important property |
| --- | --- | --- |
| GLASS\_DATA\_REPORT\_REQUEST  EFEM이 전체 실 수량을 모르는 경우(언패킹 로더에만 해당됨):. | EFEM이 카세트 매핑을 통해서 전체 실 수량을 알고 있는 경우:. | Tool에서 언로딩시 "THIS\_GLASSORDER"를 정확하게 보내 주어야 마지막 글래스인 경우 EFEM이 작업완료 진행을 할 수 있다. |
| Tool에서 언로딩시 GLASS\_DATA\_REPORT\_REQUEST의 "ULGFlag"에 "last"임을 명시해야 EFEM은 이 글래스가 마지막 글래스임을 알고 완료 진행을 할 수 있다.. |  |
| GLASS\_SCRAP\_NOTIFY\_REQUEST  . | EFEM이 카세트 매핑을 통해서 전체 실 수량을 알고 있는 경우:. | Tool에서 스크랩 발생시 "THIS\_GLASSORDER"를 정확하게 보내 주어야 마지막 글래스인 경우 EFEM이 작업완료 진행을 할 수 있다. |
| Tool에서 스크랩발생시 "ULGFlag"에 "last"임을 명시해야 EFEM은 이 글래스가 마지막 글래스임을 알고 완료 진행을 할 수 있다.. |  |

##### 2.5.2.2.2 efem-eq 인터페이스에서 사용하는 소켓 메시지 리스트

| Message Name | Direction | Description |
| --- | --- | --- |
| WORKORDER\_START\_REQUEST | Tool ← EFEM, reply | Work Order is the top/bottom total job of physical carrier batch. |
| WORKORDER\_START\_RESPONSE | Tool → EFEM | Tool은 work order 내부에 있는 ppname을 체크하고 efem에 응답한다. |
| WORKSEQPP\_START\_REQUEST | Tool ← EFEM, reply | workseqpp는 하나의 recipe를 가지고 작업하는 단위이다. Tool은 workseqpp의 recipe를 체크해야 recipe change할 시간이 필요한지 체크하고 efem에 응답한다. |
| WORKSEQPP\_START\_RESPONSE | Tool → EFEM |  |
| WORKSEQPP\_END\_REQUEST | Tool ← EFEM, reply | The end of current recipe process. |
| WORKSEQPP\_END\_RESPONSE | Tool → EFEM |  |
| MANUAL\_START\_REQUEST | Tool ← EFEM, reply | efem은 eq에 메뉴얼 모드의 시작을 알린다. |
| MANUAL\_START\_RESPONSE | Tool → EFEM |  |
| PPLIST\_REQUEST | Tool ← EFEM, reply | efem은 매뉴얼 모드에서 사용할 목적으로 eq가 가지고 있는 pplist를 요청한다. 요청 시점은 언제든지 가능해야 한다. |
| PPLIST\_RESPONSE | Tool → EFEM |  |
| GLASS\_DATA\_REPORT\_REQUEST | Tool ↔ EFEM | efem이든 eq 이든 한 장의 글래스를 로딩/언로딩 할때 상대방에게 해당 글래스의 정보를 먼저 전달해야 한다. |
| GLASS\_SCRAP\_NOTIFY\_REQUEST | Tool ↔ EFEM | efem/eq 어느 쪽이든 스크랩 상황이 발생하면 조치후 이 메시지를 전송하여 스크랩 정보를 서로 동기화 해야 한다. |
| GLASS\_SCRAP\_NOTIFY\_RESPONSE | Tool ↔ EFEM | GLASS\_SCRAP\_NOTIFY\_REQUEST 에 대한 응답 확인 |
| EQP\_STATE\_INFO | Tool → EFEM | Tool의 상황이 글래스를 받을 수 없는 상황이라서 글래스를 더 이상 투입하지 말 것을 요청함. (단 Tool 내부에 진행중인 글래스의 배출은 할 수 있음)또는 Tool이 정상적인 상황이 되었으므로 efem이 글래스 투입 진행을 할 수 있음을 다시 알림. |
| LOAD\_REQUEST\_STAGE\_NO | Tool → EFEM | eq reqeusts efem for load glass to the specific slot |
| LOAD\_RESPONSE\_STAGE\_NO | Tool ← EFEM | efem asertain the specific slot number for loading the glass |
| UNLOAD\_REQUEST\_STAGE\_NO | Tool → EFEM | eq request efem for unloading the glass from the specific slot number |
| UNLOAD\_RESPONSE\_STAGE\_NO | Tool ← EFEM | eq asertains the specific slot number for unloading the glass |

##### 2.5.2.2.3 Single/Multi Stage 공통 메시지

###### 2.5.2.2.3.1 WORKORDER\_START\_REQUEST

☞ 이것은 EFEM이 실물 케리어를 작업하기 시작할 때 전달할 것이다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM, reply |
| 설명 | Tool은 래시피 이름이 존재하지 않을 때만 에러로 회신한다. |
| Format | {  "$id": "http://efem-tool-interface/schemas/WorkOrder\_start\_request",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKORDER\_START\_REQUEST",  "type": "string"  },  "IsManual": {  "description": "If true, Tool runs in manual mode. Tool should not wait PP SELECT from EAP",  "type": "boolean"  },  "PPLIST": {  "type": "array",  "items": {  "type": "object",  "properties": {  "SEQNO": {  "type": "number",  "description": "The number what EAP has assigned for this sequence"  },  "PPID": {  "type": "string",  "description": "recipe name in the Tool"  },  "CARRIER\_SLOTS": {  "type": "array",  "items": {  "$ref": "/schemas/carrier\_slots"  }  }  },  "required": ["SEQNO", "PPID", "CARRIER\_SLOTS"]  }  }  },  "required": ["COMMAND", "IsManual", "PPLIST"]  }  {  "$id": "http://efem-tool-interface/schemas/carrier\_slots",  "type": "object",  "properties": {  "PORT\_NO": {  "type": "number",  "description": "load port number"  },  "CARRIER\_ID": {  "type": "string",  "description": "carrier id"  }  "SLOTS": {  "type": "array",  "items": {  "type": "object",  "properties": {  "SLOT\_NO: {  "type": "number",  "description": "slot number increasing base on 1 from bottom to up. Only slot for real glass existing."  },  "GLASS\_ID: {  "type": "string",  "description": "glass id"  },  "FLIPINFO: {  "type": "string",  "description": "flip information for the glass. It consists of three letters."  }  },  "required": ["SLOT\_NO", "GLASS\_ID", "FLIPINFO"]  }  }  },  "required": ["PORT\_NO", "CARRRIER\_ID", "SLOTS"]  } |
| Detail for flipinfo | first letter: glass side when carrier entering the load port  middle letter: glass side when entering the tool  last letter: glass side returning to the carrier  The glass will be flipped in the case when first letter is not equal to the second letter. |
| Sample | {  "COMMAND": "WORKORDER\_START\_REQUEST",  "PPLIST": [  {  "SEQNO": 1,  "PPID": "recipe001",  "CARRIER\_SLOTS": [  {  "PORT\_NO": 1,  "CARRIER\_ID": "Carrier0001",  "SLOTS": [  {  "SLOT\_NO": 1,  "GLASS\_ID": "Glass0001",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 2,  "GLASS\_ID": "Glass0002",  "FLIPINFO": "TTT"  },  ...  {  "SLOT\_NO": 18,  "GLASS\_ID": "Glass0018",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 19,  "GLASS\_ID": "Glass0019",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 20,  "GLASS\_ID": "Glass0020",  "FLIPINFO": "TTT"  }  ]  }  ]  },  {  "SEQNO": 2,  "PPID": "recipe002",  "CARRIER\_SLOTS": [  {  "PORT\_NO": 1,  "CARRIER\_ID": "Carrier0001",  "SLOTS": [  {  "SLOT\_NO": 1,  "GLASS\_ID": "Glass0001",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 2,  "GLASS\_ID": "Glass0002",  "FLIPINFO": "TBT"  },  ...  {  "SLOT\_NO": 18,  "GLASS\_ID": "Glass0018",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 19,  "GLASS\_ID": "Glass0019",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 20,  "GLASS\_ID": "Glass0020",  "FLIPINFO": "TBT"  }  ]  }  ]  }  ]  } |

###### 2.5.2.2.3.2 WORKORDER\_START\_RESPONSE

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | 주로 적용할 ppid가 유효한지 체크한다. |
| Format | {  "$id": "http://efem-tool-interface/schemas/WorkOrder\_start\_response",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKORDER\_START\_RESPONSE",  "type": "string"  },  "PPAckList": {  "type": "array",  "items": {  "type": "object",  "properties": {  "SEQNO": {  "type": "number",  "description": "sequential number"  },  "PPACK": {  "type": "number",  "description": "0: recipe existing, -1: no such recipe"  },  },  "required": ["SEQNO", "PPACK"]  }  }  },  "required": ["COMMAND", "PPAckList"]  } |
| Sample | {  "COMMAND": "WORKORDER\_START\_RESPONSE",  "PPAckList": [  {  "SEQNO": 1,  "PPACK": 0  },  {  "SEQNO": 2,  "PPACK": -1  }  ]  } |

###### 2.5.2.2.3.3 WORKSEQPP\_START\_REQUEST

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM, reply |
| 설명 | 원래 CARRIER\_SLOTS 가 여기에 있었으나 WORKORDER\_START\_REQUEST로 이동함. |
| Format | {  "$id": "http://efem-tool-interface/workseqpp\_start\_reqeust",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKSEQPP\_START\_REQUEST",  "type": "string"  },  "SEQNO": {  "type": "number",  "description": "sequential number identifying this workseqpp"  },  "PJID": {  "type": "string",  "description": "The process job id received from EAP."  },  "PPID": {  "type": "string",  "description": "ppid existing in the tool"  },  "APCP": {  "type": "string",  "description": "Optional. Advanced Process Control Parameters. For a large data base64 encoding can be used.",  }  },  "required": ["COMMAND", "SEQNO", "PJID", "PPID"]  } |
| Sample | {  "COMMAND": "WORKSEQPP\_START\_REQUEST",  "SEQNO": "1",  "PJID": "ProcessJob0001",  "PPID": "recipe01",  "APCP": ""  } |

###### 2.5.2.2.3.4 WORKSEQPP\_START\_RESPONSE

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | Tool은 해당 work에 대한 승락 여부를 보낸다. |
| Format | {  "$id": "http://efem-tool-interface/workseqpp\_start\_response",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKSEQPP\_START\_RESPONSE",  "type": "string"  },  "SEQNO": {  "description": "seq. no",  "type": "number"  },  "ACK": {  "description": "0: Immediately proceeding, 1: Recipe change, -1: Can't proceed for recipe mismatching, -2; Internal Error ",  "type": "number"  },  "RECIPE\_CHANGE\_TIME": {  "type": "number"  "description": "Optional. When recipe change case(1), how much time needed in seconds.",  }  },  "required": ["COMMAND", "SEQNO", "ACK"]  } |
| Sample | {  "COMMAND": "WORKSEQPP\_START\_RESPONSE",  "SEQNO": 1,  "ACK": 1,  "RECIPE\_CHANGE\_TIME": 30  } |

###### 2.5.2.2.3.5 WORKSEQPP\_END\_REQUEST

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM, reply |
| 설명 | seqno로 기존의 수신한 workseq를 식별한다. |
| Format | {  "$id": "http://efem-tool-interface/workseqpp\_end\_verify\_request",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKSEQPP\_END\_VERIFY\_REQUEST",  "type": "string"  },  "SEQNO": {  "description": "work sequnce number",  "type": "number"  }  },  "required": ["COMMAND", "SEQNO"]  } |
| Sample | {  "COMMAND": "WORKSEQPP\_END\_REQUEST",  "SEQNO": 1  } |

###### 2.5.2.2.3.6 WORKSEQPP\_END\_RESPONSE

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | 단순한 회신일 뿐이다. |
| Format | {  "$id": "http://efem-tool-interface/workseqpp\_end\_response.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "WORKSEQPP\_END\_RESPONSE",  "type": "string"  },  "SEQNO": {  "type": "number"  "description": "seq. no",  }  },  "required": ["COMMAND", "SEQNO"]  } |
| Sample | {  "COMMAND": "WORKSEQPP\_END\_RESPONSE",  "SEQNO": 1  } |

###### 2.5.2.2.3.7 PPLIST\_REQUEST

efem은 매뉴얼 모드에서 사용할 목적으로 eq가 가지고 있는 pplist를 요청한다. 요청 시점은 언제든지 가능해야 한다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM, reply |
| 설명 | WorkOrder 기반한 작업을 진행하기 위해서 EFEM은 Tool이 가지고 있는 PPList를 정확히 알 필요가 있다. |
| Format | {  "$id": "http://efem-tool-interface/pplist\_request",  "type": "object",  "properties": {  "COMMAND": {  "description": "PPLIST\_REQUEST",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "PPLIST\_REQUEST"  } |

###### 2.5.2.2.3.8 PPLIST\_RESPONSE

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | Tool은 자신이 가진 PPList를 EFEM에 알린다. |
| Format | {  "$id": "http://efem-tool-interface/pplist\_response",  "type": "object",  "properties": {  "COMMAND": {  "description": "PPLIST\_RESPONSE",  "type": "string"  },  "PPLIST": {  "description": "pplist array",  "type": "array",  "items": {  "type": "object",  "properties": {  "PPID": {  "type": "string",  "description": "name"  },  "EXPLANATION": {  "type": "string",  "description": "Explanation for ppid. It can be null"  }  },  "required": ["PPID"]  }  }  },  "required": ["COMMAND", "PPLIST"]  } |
| Sample | {  "COMMAND": "PPLIST\_REQUEST",  "PPLIST": [  {  "PPID": "pp01",  "EXPLANATION": "golden recipe"  },  {  "PPID": "pp02",  "EXPLANATION": "special recipe"  },  {  "PPID": "pp03",  "EXPLANATION": "cleaning recipe"  }  ]  } |

###### 2.5.2.2.3.9 GLASS\_DATA\_REPORT\_REQUEST

efem, tool 모두, 매 장의 글래스를 로딩/언로딩 할때 상대방에게 해당 글래스의 정보를 먼저 전달해야 한다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ↔ EFEM |
| 설명 | 공정이 완료되지 않고 언로딩된 글래스는 PROCESS\_STATE=0 으로 셋팅하고 어느 공정까지 완료되었는지를 FINISHED\_SUBSTEPS에 표시한다. 민감한 정보의 보호를 위해서 FINISHED\_SUBSTEPS는 어떤 공정인지를 인지할 수 있을 정도로만 표시해도 된다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/WORK-VERIFY.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "GLASS\_DATA\_REPORT\_REQUEST",  "type": "string"  },  "HANDOVER\_PORT": {  "description": "Hand over port id for multi port tool, generally it is port1",  "type": "string"  },  "GLASS\_INFO": {  "type": "object",  "properties": {  "CARRIER\_ID": {  "type": "string",  "description": "The carrier id from where the glass is withdrawn"  },  "PORT\_ID": {  "type": "number",  "description": "The port id from where the glass is withdrawn. This number is the same as what the Host recognize."  },  "LOT\_ID": {  "type": "string",  "description": "..."  },  "GLASS\_ID": {  "type": "string"  }  "SLOT\_NO": {  "type": "number",  "description": "1 based number increasing from bottom of the carrier"  },  "MAPPED\_GLASSCOUNT": {  "type": "number",  "description": "The slot mapped total existing glass count in the carrier."  },  "THIS\_GLASSORDER": {  "type": "number",  "description": "1 based serial number for this glass including scrapped glass."  },  "MANUAL\_INSP": {  "type": "Boolean",  "description": "true: This glass will be visually inspected in the flipper, or in the measurement room. false: No manual inspection"  },  "USAGE": {  "type": "string",  "description": "Product", "Dry"  },  "PROCESS\_STATE": {  "type": "number",  "description": "0: undone, 1: done. When EFEM send glass info. this should be 0, when Tool send glass info. this will be 1"  },  "FINISHED\_SUBSTEPS": {  "type": "string",  "description": "This is valid if the PROCESS\_STATE is 0. Indicate upto which process is completed. ex) if final substep is step5 and this glass is processed upto step3 and unloaded, /step1/step2/step3"  },  "FLIP\_INFO": {  "type": "string",  "description": "The same as in WorkOrder CarrierSlot"  },  "PPID": {  "type": "string",  "description": "Recipe ID"  },  "ULGFlag": {  "type": "string",  "description": "first/middle/last. It's applicable only for the special Tool (Unpacking Loader). For other Tool, Please ignore this property."  }  },  "required": [  "CARRIER\_ID",  "PORT\_ID",  "LOT\_ID",  "GLASS\_ID",  "SLOT\_NO",  "MAPPED\_GLASS\_COUNT",  "THIS\_GLASSORDER",  "MANUAL\_INSP",  "USAGE",  "PROCESS\_STATE",  "FLIP\_INFO",  "PPID"  ]  },  "required": [  "COMMAND",  "HANDOVER\_PORT",  "GLASS\_INFO"  ]  }  } |
| Sample | {  "COMMAND": "GLASS\_DATA\_REPORT\_REQUEST",  "HANDOVER\_PORT": "port1",  "GLASS\_INFO": {  "CARRIER\_ID": "Carrier0001",  "PORT\_ID": 1,  "LOT\_ID": "Lot0001",  "GLASS\_ID": "Glass001",  "SLOT\_NO": 1,  "MAPPED\_GLASSCOUNT": 20,  "THIS\_GLASSORDER": 1,  "MANUAL\_INSP": true,  "USAGE": "Dry",  "PROCESS\_STATE": 0,  “FINISHED\_SUBSTEPS”: “/step1/step2”,  "FLIP\_INFO": "TTT",  "PPID": "Recipe001",  "ULGFlag": "last"  }  } |

###### 2.5.2.2.3.10 GLASS\_DATA\_REPORT\_RESPONSE

GLASS\_DATA\_REPORT\_REQUEST 메시지에 대한 회신.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ↔ EFEM |
| 설명 | GALSS\_DATA\_REPORT\_REQUEST 메시지의 전달은 중요하므로 회신을 추가한다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/WORK-VERIFY.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "GLASS\_DATA\_REPORT\_RESPONSE",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "GLASS\_DATA\_REPORT\_RESPONSE"  } |

###### 2.4.2.2.3.11 GLASS\_SCRAP\_NOTIFY\_REQUEST

efem/eq 어느 쪽이든 스크랩 상황이 발생하면 조치후 이 메시지를 전송하여 스크랩 정보를 서로 동기화 해야 한다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ↔ EFEM |
| 설명 |  |
| Format | {  "$id": "http://efem-tool-interface/glass\_scrap\_request.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "GLASS\_SCRAP\_NOTIFY\_REQUEST",  "type": "string"  },  "GLASS\_INFO": {  "type": "object",  "properties": {  "CARRIER\_ID": {  "type": "string"  },  "PORT\_ID": {  "type": "number"  },  "LOT\_ID": {  "type": "string",  "description": "..."  },  "GLASS\_ID": {  "type": "string"  },  "SLOT\_NO": {  "type": "number",  "description": "1 based number increasing from bottom of the carrier"  },  "THIS\_GLASSORDER": {  "type": "number",  "description": "1 based serial number for this glass including scrapped glass. What is in the GLASS\_DATA\_REPORT\_REQUEST message."  },  "MAPPED\_GLASSCOUNT": {  "type": "number",  "description": "The slot mapped total existing glass count in the carrier. What is in the GLASS\_DATA\_REPORT\_REQUEST message."  }  },  "required": ["CARRIER\_ID", "PORT\_ID", "LOT\_ID", "GLASS\_ID", "SLOT\_NO", "THIS\_GLASSORDER"]  }  },  "required": ["COMMAND", "GLASS\_INFO"]  } |
| Sample | {  "COMMAND": "GLASS\_SCRAP\_NOTIFY\_REQUEST",  "GLASS\_INFO": {  "CARRIER\_ID": "Carrier0001",  "PORT\_ID": 1,  "LOT\_ID": "Lot0001",  "GLASS\_ID": "Glass001",  "SLOT\_NO": 1,  "THIS\_GLASSORDER": 19,  "MAPPED\_GLASSCOUNT": 20,  }  } |

###### 2.5.2.2.3.12 GLASS\_SCRAP\_NOTIFY\_RESPONSE

GLASS\_SCRAP\_NOTIFY\_REQUEST 에 대한 응답 확인

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ↔ EFEM |
| 설명 |  |
| Format | {  "$id": "http://efem-tool-interface/glass\_scrap\_notify\_response",  "type": "object",  "properties": {  "COMMAND": {  "description": "GLASS\_SCRAP\_NOTIFY\_RESPONSE",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "GLASS\_SCRAP\_NOTIFY\_RESPONSE"  } |

###### 2.5.2.2.3.13 EQP\_STATE\_INFO

2.5.2.2.3.13.1 이 메시지의 용도

Tool의 상황이 글래스를 받을 수 없는 상황이라서 글래스를 더 이상 투입하지 말 것을 요청함. (단 Tool 내부에 진행중인 글래스의 배출은 할 수 있음)

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | EFEM은 S5F1로 Host에 보고하고, 경광등은 yellow 로 한다. 이 경우에도 글래스 배출은 할 수 있어야 한다. |
| Format | {  "$id": "http://efem-tool-interface/eqp\_state\_info.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "EQP\_STATE\_INFO",  "type": "string"  },  "ENABLE": {  "type": "number",  "description": "0: diable, 1: enable",  }  },  "required": ["COMMAND", "ENABLE"]  } |
| Sample | {  "COMMAND": "EQP\_STATE\_INFO",  "ENABLE": 0  } |

###### 2.5.2.2.3.14 ABORT\_REQUEST

EAP에서 Abort를 요청했을 때 EFEM이 Tool에 이 메시지를 보낸다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM |
| 설명 | EFEM은 EAP로부터 CJAbort 메시지를 받으면 ABORT\_REQUEST 메시지를 Tool에 전송한다.  Tool은 즉각 공정을 중단하는 절차를 수행하고 내부에 들어있는 글래스를 언로딩해야 한다. |
| Format | {  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_REQUEST",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "ABORT\_REQUEST",  } |

###### 2.5.2.2.3.15 ABORT\_REQUEST\_ACK

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | PROCESS\_ABORT\_REQUEST에 대한 Tool의 회신 (실제로 Abort를 완료했다는 의미는 아님) |
| Format | {  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_REQUEST\_ACK",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "PROCESS\_ABORT\_RESPONSE"  } |

###### 2.5.2.2.3.16 ABORT\_STARTED

Tool에서 Abort를 시작했을 경우 EFEM에 Abort가 시작되었음을 알려야 한다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 |  |
| Format | {  "$id": "http://efem-tool-interface/workorder\_abort\_request.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_STARTED",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "ABORT\_STARTED",  } |

###### 2.5.2.2.3.17 ABORT\_STARTED\_ACK

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM |
| 설명 |  |
| Format | {  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_STARTED\_ACK",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "PROCESS\_ABORT\_RESPONSE"  } |

###### 2.5.2.2.3.18 ABORT\_COMPLETED

Tool에 있는 모든 글래스를 언로딩 시퀀스를 통해서든, 사용자가 수동으로 제거하든, 모든 글래스가 Tool에서 제거되면 Tool은 이 메시지를 EFEM에 전송한다.

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 |  |
| Format | {  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_COMPLETED",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "ABORT\_COMPLETED"  } |

###### 2.5.2.2.3.19 ABORT\_COMPLETED\_ACK

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM |
| 설명 | ABORT\_COMPLETED에 대한 EFEM의 회신. |
| Format | {  "type": "object",  "properties": {  "COMMAND": {  "description": "ABORT\_COMPLETED\_ACK",  "type": "string"  }  },  "required": ["COMMAND"]  } |
| Sample | {  "COMMAND": "ABORT\_COMPLETED\_ACK"  } |

##### 2.5.2.2.4 Multi Stage 전용 메시지

아래 메시지들은 multi stage Tool 에만 해당한다.

###### 2.5.2.2.4.1 LOAD\_REQUEST\_STAGE\_NO

eq reqeusts efem for load glass to the specific slot

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | Tool은 EFEM에 몇 번 stage에 글래스를 넣으라고 요청한다. 만약 stage number가 0 이면 어떤 stage 든 접근하지 말것을 요청하는 것이다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/xxx.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "LOAD\_REQUEST\_STAGE\_NO",  "type": "string"  },  "STAGE\_NO": {  "description": "Requested stage number to load glass. If it is 0, Tool reqests EFEM no to access all stage.",  "type": "Number"  },  "HANDOVER\_PORT": {  "description": "port1 or port2, Generally it is port1",  "type": "string"  },  "GLASS\_ID": {  "description": "The glass id what the EFEM requested to load.",  "type": "string"  }  },  "required": ["COMMAND", "STAGE\_NO", "HANDOVER\_PORT", "GLASS\_ID"]  } |
| Sample | {  "COMMAND": "LOAD\_REQUEST\_STAGE\_NO",  "STAGE\_NO": "1",  "HANDOVER\_PORT": "port1",  "GLASS\_ID": "glass1"  } |

###### 2.5.2.2.4.2 LOAD\_RESPONSE\_STAGE\_NO

efem asertain the specific slot number for loading the glass

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM |
| 설명 | EFEM은 Tool이 요청한 Stage에 글래스를 넣을 것을 알린다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/xxx.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "LOAD\_RESPONSE\_STAGE\_NO",  "type": "string"  },  "HANDOVER\_PORT": {  "description": "Hand over port id for multi port tool, generally it is port1 ",  "type": "string"  },  "STAGE\_NO": {  "description": "1 based stage number top down direction",  "type": "number"  },  "GLASS\_ID": {  "description": "The glass id what the EFEM requested to load.",  "type": "string"  }  },  "required": ["COMMAND", "HANDOVER\_PORT", "STAGE\_NO", "GLASS\_ID"]  } |
| Sample | {  "COMMAND": "LOAD\_RESPONSE\_STAGE\_NO",  "HANDOVER\_PORT": "port1",  "STAGE\_NO": 10,  "GLASS\_ID": 'glass1'  } |

###### 2.5.2.2.4.3 UNLOAD\_REQUEST\_STAGE\_NO

eq request efem for unloading the glass from the specific slot number

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool → EFEM |
| 설명 | Tool은 해당 stage로 부터 글래스를 언로딩할 것을 EFEM에게 요청한다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/xxx.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "UNLOAD\_REQUEST\_STAGE\_NO",  "type": "string"  },  "HANDOVER\_PORT": {  "description": "The number of port where glass is transferred. If this is not assigned it should be port1",  "type": "string"  },  "STAGE\_NO": {  "description": "Tool's stage number from which glass should be unloaded",  "type": "number"  }  },  "required": ["COMMAND", "HANDOVER\_PORT", "STAGE\_NO"]  } |
| Sample | {  "COMMAND": "UNLOAD\_REQUEST\_STAGE\_NO",  "HANDOVER\_PORT":"port1",  "STAGE\_NO": "5"  } |

###### 2.5.2.2.4.4 UNLOAD\_RESPONSE\_STAGE\_NO

eq asertains the specific slot number for unloading the glass

| 항목 | 내용 |
| --- | --- |
| 전송 방향 | Tool ← EFEM |
| 설명 | EFEM 은 해당 stage no. 에서 글래스를 뺄 것을 확인한다. |
| Format | {  "$schema": "https://json-schema.org/draft/2020-12/schema",  "$id": "http://efem-tool-interface/xxx.json",  "type": "object",  "properties": {  "COMMAND": {  "description": "UNLOAD\_RESPONSE\_STAGE\_NO",  "type": "string"  },  "HANDOVER\_PORT": {  "description": "The number of port where glass is transferred. If this is not assigned it should be port1",  "type": "string"  },  "STAGE\_NO": {  "description": "1 based stage number top down direction",  "type": "number"  }  },  "required": ["COMMAND", "HANDOVER\_PORT", "STAGE\_NO"]  } |
| Sample | {  "COMMAND": "UNLOAD\_RESPONSE\_STAGE\_NO",  "HANDOVER\_PORT":"port1",  "STAGE\_NO": 5  } |

### 2.5.3 Socket Message Samples

이 샘플들은 참고용으로 각 소켓 메시지 설명에 있는 것과 동일한 것을 한 곳에 모은 것이다.

| Message | Sample |
| --- | --- |
| WORKORDER\_START\_REQUEST | {  "COMMAND": "WORKORDER\_START\_REQUEST",  "PPLIST": [  {  "SEQNO": 1,  "PPID": "recipe001",  "CARRIER\_SLOTS": [  {  "PORT\_NO": 1,  "CARRIER\_ID": "Carrier0001",  "SLOTS": [  {  "SLOT\_NO": 1,  "GLASS\_ID": "Glass0001",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 2,  "GLASS\_ID": "Glass0002",  "FLIPINFO": "TTT"  },  …,  {  "SLOT\_NO": 18,  "GLASS\_ID": "Glass0018",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 19,  "GLASS\_ID": "Glass0019",  "FLIPINFO": "TTT"  },  {  "SLOT\_NO": 20,  "GLASS\_ID": "Glass0020",  "FLIPINFO": "TTT"  }  ]  }  ]  },  {  "SEQNO": 2,  "PPID": "recipe002",  "CARRIER\_SLOTS": [  {  "PORT\_NO": 1,  "CARRIER\_ID": "Carrier0001",  "SLOTS": [  {  "SLOT\_NO": 1,  "GLASS\_ID": "Glass0001",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 2,  "GLASS\_ID": "Glass0002",  "FLIPINFO": "TBT"  },  …,  {  "SLOT\_NO": 18,  "GLASS\_ID": "Glass0018",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 19,  "GLASS\_ID": "Glass0019",  "FLIPINFO": "TBT"  },  {  "SLOT\_NO": 20,  "GLASS\_ID": "Glass0020",  "FLIPINFO": "TBT"  },  ]  }  ]  }  ]  } |
| WORKORDER\_START\_RESPONSE | {  "COMMAND": "WORKORDER\_START\_RESPONSE",  "PPAckList": [  {  "SEQNO": 1,  "PPACK": 0  },  {  "SEQNO": 2,  "PPACK": -1  }  ]  } |
| WORKSEQPP\_START\_REQUEST | {  "COMMAND": "WORKSEQPP\_START\_REQUEST",  "PPID": "Recipe001",  "APCP": ""  } |
| WORKSEQPP\_START\_RESPONSE | {  "COMMAND": "WORKSEQPP\_START\_RESPONSE",  "SEQNO": 1,  "ACK": 1,  "RECIPE\_CHANGE\_TIME": 30  } |
| WORKSEQPP\_END\_REQUEST | {  "COMMAND": "WORKSEQPP\_END\_REQUEST",  "SEQNO": 1  } |
| WORKSEQPP\_END\_RESPONSE | {  "COMMAND": "WORKSEQPP\_END\_RESPONSE",  "SEQNO": 1  } |
| MANUAL\_START\_REQUEST | {  "COMMAND": "MANUAL\_START\_REQUEST"  } |
| MANUAL\_START\_RESPONSE | {  "COMMAND": "MANUAL\_START\_RESPONSE",  "ACK": 0  } |
| PPLIST\_REQUEST | {  "COMMAND": "PPLIST\_REQUEST"  } |
| PPLIST\_RESPONSE | {  "COMMAND": "PPLIST\_REQUEST",  "PPLIST": [  {  "PPID": "pp01",  "EXPLANATION": "golden recipe"  },  {  "PPID": "pp02",  "EXPLANATION": "special recipe"  },  {  "PPID": "pp03",  "EXPLANATION": "cleaning recipe"  }  ]  } |
| GLASS\_DATA\_REPORT\_REQUEST | {  "COMMAND": "GLASS\_DATA\_REPORT\_REQUEST",  "HANDOVER\_PORT": "port1",  "GLASS\_INFO": {  "CARRIER\_ID": "Carrier0001",  "PORT\_ID": 1,  "LOT\_ID": "Lot0001",  "GLASS\_ID": "Glass001",  "SLOT\_NO": 1,  "MAPPED\_GLASSCOUNT": 20,  "THIS\_GLASSORDER": 1,  "USAGE": "Dry",  "PROCESS\_STATE": 0,  "FLIP\_INFO": "TTT"  }  } |
| GLASS\_SCRAP\_NOTIFY\_REQUEST | {  "COMMAND": "GLASS\_SCRAP\_NOTIFY\_REQUEST",  "GLASS\_INFO": {  "CARRIER\_ID": "Carrier0001",  "PORT\_ID": 1,  "LOT\_ID": "Lot0001",  "GLASS\_ID": "Glass001",  "SLOT\_NO": 1  }  } |
| GLASS\_SCRAP\_NOTIFY\_RESPONSE | {  "COMMAND": "GLASS\_SCRAP\_NOTIFY\_RESPONSE"  } |
| EQP\_STATE\_INFO | {  "COMMAND": "EQP\_STATE\_INFO",  "ENABLE": 0,  "ALARM\_CODE": 0,  "ALARM\_DESC": "Bath#2 is in abnormal condition"  } |
| PIO\_RESTART\_REQUEST | {  "COMMAND": "PIO\_RESTART\_REQUEST",  "STAGE": "load\_complete\_confirm\_timeout",  } |
| LOAD\_REQUEST\_STAGE\_NO | {  "COMMAND": "LOAD\_REQUEST\_STAGE\_NO",  "STAGE\_NO": "1",  "HANDOVER\_PORT": "port1",  "GLASS\_ID": "glass1"  } |
| LOAD\_RESPONSE\_STAGE\_NO | {  "COMMAND": "LOAD\_RESPONSE\_STAGE\_NO",  "HANDOVER\_PORT": "port1",  "STAGE\_NO": 10,  "GLASS\_ID": 'glass1'  } |
| UNLOAD\_REQUEST\_STAGE\_NO | {  "COMMAND": "UNLOAD\_REQUEST\_STAGE\_NO",  "HANDOVER\_PORT":"port1",  "STAGE\_NO": "5"  } |
| UNLOAD\_RESPONSE\_STAGE\_NO | {  "COMMAND": "UNLOAD\_RESPONSE\_STAGE\_NO",  "HANDOVER\_PORT":"port1",  "STAGE\_NO": 5  } |

# 3 Interface/Integration Test plan

## 3.1 Integration Simulation Test before shipment

| 항목 | 내용 |
| --- | --- |
| 테스트 장소 | 한화 모멘텀 아산 사업장 |
| 테스트 일시 | 2023년 입고 전 |
| 테스트 방법 | 1. Host-EFEM-Tool 인테그레이션 하여 Host로 부터 Job 내려서 EFEM/Tool이 각자 Job 수신하고 EFEM이 글래스 투입하고 Tool이 글래스 가공 후 반환하는 시나리오를 시뮬레이션으로 수행.  2. EFEM/Tool 공급사는 PIO를 input/output 테스트 할 수 있는 보드를 준비해야 한다.  3. PIO 테스트는 배선 연결된 상태에서 개발한 소프트웨어를 이용해서 먼저 시그널별로 서로 확인하고, PIO 케이스별로 확인할 수 있어야 한다. |

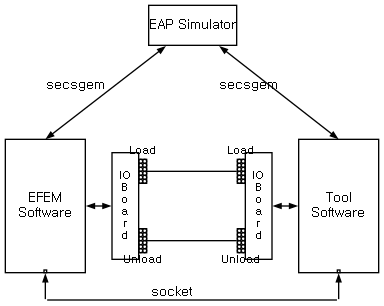


Figure 1 Simulation Test layout

## 3.2 현장 셋업 후 Dry Run 테스트:

| 항목 | 내용 |
| --- | --- |
| 테스트 장소 | 앱솔릭스 현지 |
| 테스트 일시 | 셋업 후 |
| 테스트 방법 | 실물 글래스를 투입하나 실제 글래스에 물리적/화학적 공정은 하지 않고 핸들링 위주로 테스트.[GLASS\_DATA\_REPORT\_REQUEST](#bm_GLASS_DATA_REPORT_REQUEST)  메시지의 USAGE를 Dry로 셋팅하는 것으로 Dry Run 을 구별할 예정 |

