목차

Zone Control API 1
안티패스백 구역 1
시간 지정 안티패스백 구역 1
화재 경보 구역 2
스케줄 잠금/개방 구역 2
경비 경보 구역 2
인터락 구역 3
Ethernet 구역 3
Lift스케줄 잠금/개방 구역 4
콜백 함수 4
OnCheckGlobalAPBViolation 4
OnCheckGlobalAPBViolationByDoorOpen 5
OnUpdateGlobalAPBViolationByDoorOpen 5
구조체 6
BS2ZoneStatus 6
BS2ApbMember 6
BS2TimedApbMember 7
BS2FireSensor 7
BS2AntiPassbackZone 7
BS2TimedAntiPassbackZone 9
BS2FireAlarmZone 10
BS2ScheduledLockUnlockZone 11
BS2IntrusionAlarmZone 12
BS2IntrusionAlarmZoneBlob 14
BS2InterlockZone 14
BS2InterlockZoneBlob 15
BS2DeviceZoneEntranceLimitMaster 15
BS2DeviceZoneEntranceLimitMember 17
BS2DeviceZoneFireAlarmMaster 18
BS2DeviceZoneFireAlarmMember 18
BS2DeviceZoneFireAlarmMemberInfo 19
BS2DeviceZoneFireSensor 19
BS2DeviceZone 20
BS2DeviceZoneAGEntranceLimit 20
BS2DeviceZoneMasterConfig 21
BS2LiftFloors 22
BS2LiftLockUnlockZone 22

Zone Control API

하나의 관리 지역을 특정 목적에 맞게 여러 구역으로 나눠 출입을 통제할 수 있습니다. 구역을 설정하면 출입 인증 장치, 출입문의 동작을 제어할 수 있습니다.

안티패스백 구역

사용자가 자신의 카드를 제삼자에게 빌려준다든지 자신의 지문을 이용하여 제삼자를 안으로 들이는 상황을 방지하려면, 이중출입 방지 구역을 사용합니다. 이 구역은 두 가지 종류의 옵션(소프트, 하드)을 지원합니다. 소프트를 선택하면, 안티패스백을 위반해도 출입을 허용하지만 위반 사실을 사용자 로그에 기록합니다. 하드를 선택하면 모든 안티패스백을 허락하지 않으며 위반 사실을 사용자 로그에 기록합니다.

- BS2_GetAntiPassbackZone: 주어진 안티패스백 구역을 가져옵니다.
- BS2_GetAllAntiPassbackZone: 모든 안티패스백 구역을 가져옵니다.
- BS2_GetAntiPassbackZoneStatus: 주어진 안티패스백 구역의 상태 정보를 가져옵니다.
- BS2_GetAllAntiPassbackZoneStatus: 모든 안티패스백 구역의 상태 정보를 가져옵니다.
- BS2_SetAntiPassbackZone: 안티패스백 구역을 설정합니다.
- BS2_SetAntiPassbackZoneAlarm: 안티패스백 구역의 경보 상태를 갱신합니다.
- BS2_RemoveAntiPassbackZone: 주어진 안티패스백 구역을 제거합니다.
- BS2_RemoveAllAntiPassbackZone: 모든 안티패스백 구역을 제거합니다.
- BS2_ClearAntiPassbackZoneStatus: 주어진 사용자를 안티패스백 구역의 규칙을 위반하지 않은 상태로 갱신합니다.
- BS2_ClearAllAntiPassbackZoneStatus: 모든 사용자를 안티패스백 구역의 규칙을 위반하지 않은 상태로 갱신합니다.
- BS2_SetCheckGlobalAPBViolationHandler: 안티패스백 경보가 발생했을 때 글로벌 판정을 위해 콜백 함수를 등록합니다.
- BS2_CheckGlobalAPBViolation: 글로벌 안티패스백 판정 결과를 장치로 전송합니다.
- BS2_SetGlobalAPBViolationByDoorOpenHandler: [+ 2.7.0] 출입문 센서 기반의 안티패스백 경보가 발생했을 때 글로벌 판정을 위해 콜백 함수를 등록합니다.
- BS2_CheckGlobalAPBViolationByDoorOpen: [+ 2.7.0] 출입문 센서 기반의 글로벌 안티패스백 판정 결과를 장치로 전송합니다.

시간 지정 안티패스백 구역

사용자가 일정 시간내에 재진입하는 것을 막으려면, 인증 제한 구역을 사용합니다. 이 구역은 안티패스백 구역과 동일하게 두 가지 종류의 옵션(소프트, 하드)을 지원합니다.

- BS2_GetTimedAntiPassbackZone: 주어진 시간 지정 안티패스백 구역을 가져옵니다.
- BS2_GetAllTimedAntiPassbackZone: 모든 시간 지정 안티패스백 구역을 가져옵니다.
- BS2_GetTimedAntiPassbackZoneStatus: 주어진 시간 지정 안티패스백 구역의 상태 정보를 가져옵니다.
- BS2_GetAllTimedAntiPassbackZoneStatus: 모든 시간 지정 안티패스백 구역의 상태 정보를 가져옵니다.
- BS2_SetTimedAntiPassbackZone: 시간 지정 안티패스백 구역을 설정합니다.
- BS2_SetTimedAntiPassbackZoneAlarm: 시간 지정 안티패스백 구역의 경보 상태를 갱신합니다.
- BS2_RemoveTimedAntiPassbackZone: 주어진 시간 지정 안티패스백 구역을 제거합니다.
- BS2_RemoveAllTimedAntiPassbackZone: 모든 시간 지정 안티패스백 구역을 제거합니다.

- BS2_ClearTimedAntiPassbackZoneStatus: 주어진 사용자를 시간 지정 안티패스백 구역의 규칙을 위반하지 않은 상태로 갱신합니다.
- BS2_ClearAllTimedAntiPassbackZoneStatus: 모든 사용자를 시간 지정 안티패스백 구역의 규칙을 위반하지 않은 상태로 갱신합니다.

화재 경보 구역

출입 통제 구역에서 화재를 감지하고 경보를 제어하려면 화재 경보 구역을 사용합니다. 외부 입력 신호가 BioStar 시스템에 전달되면 BioStar는 자동으로 출입문을 개방하며 미리 정의한 경보를 수행합니다.

- BS2_GetFireAlarmZone: 주어진 화재 경보 구역을 가져옵니다.
- BS2_GetAllFireAlarmZone: 모든 화재 경보 구역을 가져옵니다.의
- BS2_GetFireAlarmZoneStatus: 주어진 화재 경보 구역의 상태 정보를 가져옵니다.
- BS2_GetAllFireAlarmZoneStatus: 모든 화재 경보 구역의 상태 정보를 가져옵니다.
- BS2_SetFireAlarmZone: 화재 경보 구역을 설정합니다.
- BS2_SetFireAlarmZoneAlarm: 화재 경보 구역의 경보 상태를 갱신합니다.
- BS2_RemoveFireAlarmZone: 주어진 화재 경보 구역을 제거합니다.
- BS2_RemoveAllFireAlarmZone: 모든 화재 경보 구역을 제거합니다.

스케줄 잠금/개방 구역

시간에 따라 구역을 개방하거나 폐쇄하려면 스케줄 잠금/개방 구역을 사용합니다. 이 구역은 일정 시간동안 모든 출입문을 개방하는 정시간 개방 상태와, 일정 시간동안 모든 출입문을 폐쇄하는 정시간 폐쇄 상태로 배타적으로 동작합니다.

- BS2_GetScheduledLockUnlockZone: 주어진 스케줄 잠금/개방 구역을 가져옵니다.
- BS2_GetAllScheduledLockUnlockZone: 모든 스케줄 잠금/개방 구역을 가져옵니다.
- BS2_GetScheduledLockUnlockZoneStatus: 주어진 스케줄 잠금/개방 구역의 상태 정보를 가져옵니다.
- BS2_GetAllScheduledLockUnlockZoneStatus: 모든 스케줄 잠금/개방 구역의 상태 정보를 가져옵니다.
- BS2_SetScheduledLockUnlockZone: 스케줄 잠금/개방 구역을 설정합니다.
- BS2_SetScheduledLockUnlockZoneAlarm: 스케줄 잠금/개방 구역의 경보 상태를 갱신합니다.
- BS2_RemoveScheduledLockUnlockZone: 주어진 스케줄 잠금/개방 구역을 제거합니다.
- BS2_RemoveAllScheduledLockUnlockZone: 모든 스케줄 잠금/개방 구역을 제거합니다.

경비 경보 구역

경비 구역에서 침입이나 이상 신호가 발생할 경우 신호를 수신하여 대처하는 시스템으로 사용합니다. 외부 입력 신호가 BioStar 시스템에 전달되면 BioStar는 자동으로 미리 정의한 경보를 수행합니다.

- BS2_GetIntrusionAlarmZone: 주어진 경비 경보 구역을 가져옵니다.
- BS2_GetIntrusionAlarmZoneStatus: 주어진 경비 경보 구역의 상태 정보를 가져옵니다.
- BS2_GetAllIntrusionAlarmZoneStatus: 모든 경비 경보 구역의 상태 정보를 가져옵니다.
- BS2_SetIntrusionAlarmZone: 경비 경보 구역을 설정합니다.
- BS2_SetIntrusionAlarmZoneAlarm: 경비 경보 구역의 경보 상태를 갱신합니다.

- BS2_RemoveIntrusionAlarmZone: 주어진 경비 경보 구역을 제거합니다.
- BS2_RemoveAllIntrusionAlarmZone: 모든 경비 경보 구역을 제거합니다.
- BS2_SetIntrusionAlarmZoneArm: 경비 경보 구역 상태를 경비/경비해제를 설정합니다.

인터락 구역

[CoreStation] 이중 문으로 구성되어 다른 한 쪽 문을 통과하기 전에 반드시 반대 쪽이 닫혀 있어야 하는 경우 사용됩니다.
현재 CoreStation 만 이 기능을 지원하고 있습니다.

- BS2_GetInterlockZone: 주어진 인터락 구역을 가져옵니다.
- BS2_GetInterlockZoneStatus: 주어진 인터락 구역의 상태 정보를 가져옵니다.
- BS2_GetAllInterlockZoneStatus: 모든 인터락 구역의 상태 정보를 가져옵니다.
- BS2_SetInterlockZone: 인터락 구역을 설정합니다.
- BS2_SetInterlockZoneAlarm: 인터락 구역의 경보 상태를 갱신합니다.
- BS2_RemoveInterlockZone: 주어진 인터락 구역을 제거합니다.
- BS2_RemoveAllInterlockZone: 모든 인터락 구역을 제거합니다.

Ethernet 구역

Zone Master 역할을 BioStar V2.x 서버가 아닌 특정 장치가 하며 장치 사이(Master ↔ Member)의 통신 방식으로 Ethernet TCP 통신 방식을 사용하는 Zone입니다. 현재는 기존 1.x Entrance Limit, Fire Alarm Zone에 기능만을 지원합니다. A2(FW 1.4.0 이상), BS2(FW 1.5.0 이상) and P2(FW 1.0.0이상)에서 지원합니다.

- BS2_GetDeviceZone: 주어진 Ethernet 구역을 가져옵니다.
- BS2_GetAllDeviceZone: 모든 Ethernet을 가져옵니다.
- BS2_SetDeviceZone: Ethernet 구역을 설정합니다.
- BS2_RemoveDeviceZone: 주어진 Ethernet 구역을 제거합니다.
- BS2_RemoveAllDeviceZone: 모든 Ethernet 구역을 제거합니다.
- BS2_SetDeviceZoneAlarm: Ethernet 구역의 경보 상태를 갱신합니다.
- BS2_ClearDeviceZoneAccessRecord: 주어진 사용자를 Ethernet 구역의 규칙을 위반하지 않은 상태로 갱신합니다.
- BS2_ClearAllDeviceZoneAccessRecord: 모든 사용자를 Ethernet 구역의 규칙을 위반하지 않은 상태로 갱신합니다.
- BS2_GetDeviceZoneAGEntranceLimit: 주어진 Ethernet Access Group 인증 제한 구역을 가져옵니다.
- BS2_GetAllDeviceZoneAGEntranceLimit: 모든 Ethernet Access Group 인증 제한 구역을 가져옵니다.
- BS2_SetDeviceZoneAGEntranceLimit: Ethernet Access Group 인증 제한 구역을 설정합니다.
- BS2_RemoveDeviceZoneAGEntranceLimit: 주어진 Ethernet Access Group 인증 제한 구역을 제거합니다.
- BS2_RemoveAllDeviceZoneAGEntranceLimit: 모든 Ethernet Access Group 인증 제한 구역을 제거합니다.

- BS2_GetDeviceZoneMasterConfig: Ethernet 구역의 master 장치 설정을 가져옵니다.
- BS2_SetDeviceZoneMasterConfig: Ethernet 구역의 master 장치 설정을 저장합니다.
- BS2_RemoveDeviceZoneMasterConfig: Ethernet 구역의 master 장치 설정을 제거합니다.

Lift스케줄 잠금/개방 구역

[+ 2.7.0] Elevator 제어관련, 인증권한과 무관하게 시간대를 기반으로 특정 층에 접근할 수 있도록 하는 새로운 함수들을 제공합니다.

- BS2_GetLiftLockUnlockZone: 주어진 Lift스케줄 잠금/개방 구역을 가져옵니다.
- BS2_GetAllLiftLockUnlockZone: 모든 Lift스케줄 잠금/개방 구역을 가져옵니다.
- BS2_GetLiftLockUnlockZoneStatus: 주어진 Lift스케줄 잠금/개방 구역의 상태 정보를 가져옵니다.
- BS2_GetAllLiftLockUnlockZoneStatus: 모든 Lift스케줄 잠금/개방 구역의 상태 정보를 가져옵니다.
- BS2_SetLiftLockUnlockZone: Lift스케줄 잠금/개방 구역을 설정합니다.
- BS2_SetLiftLockUnlockZoneAlarm: Lift스케줄 잠금/개방 구역의 경보 상태를 갱신합니다.
- BS2_RemoveLiftLockUnlockZone: 주어진 Lift스케줄 잠금/개방 구역을 제거합니다.
- BS2_RemoveAllLiftLockUnlockZone: 모든 Lift스케줄 잠금/개방 구역을 제거합니다.

콜백 함수

OnCheckGlobalAPBViolation

안티패스백 경보가 발생했을 때 글로벌 판정을 위한 콜백 함수입니다.

```
typedef void (*OnCheckGlobalAPBViolation)(uint32_t deviceId, uint16_t seq,
const char* userID_1, const char* userID_2, bool isDualAuth);
```

1. *deviceId*
장치 식별자입니다.

2. *seq*
패킷 일련번호입니다.

3. *userID_1*
첫 번째 사용자 ID 입니다.

4. *userID_2*
두 번째 사용자 ID 입니다.

5. *isDualAuth*
이중인증 여부를 의미합니다.

OnCheckGlobalAPBViolationByDoorOpen

출입문 센서 기반의 안티패스백 경보가 발생했을 때, 인증정보를 바탕으로 글로벌 판정을 요청하는 1차 콜백 함수입니다.

```
typedef void (*OnCheckGlobalAPBViolationByDoorOpen)(uint32_t deviceId,
uint16_t seq, const char* userID_1, const char* userID_2, bool isDualAuth);
```

1. *deviceId*
장치 식별자입니다.

2. *seq*
패킷 일련번호입니다.

3. *userID_1*
첫 번째 사용자 ID 입니다.

4. *userID_2*
두 번째 사용자 ID 입니다.

5. *isDualAuth*
이중인증 여부를 의미합니다.

OnUpdateGlobalAPBViolationByDoorOpen

출입문 센서 기반의 안티패스백 경보가 발생했을 때, 최종 사용자 출입 상태의 업데이트를 요청하는 2차 콜백 함수입니다.

```
typedef void (*OnUpdateGlobalAPBViolationByDoorOpen)(uint32_t deviceId,
uint16_t seq, const char* userID_1, const char* userID_2, bool isDualAuth);
```

1. *deviceId*
장치 식별자입니다.

2. *seq*
패킷 일련번호입니다.

3. *userID_1*
첫 번째 사용자 ID 입니다.

4. *userID_2*
두 번째 사용자 ID 입니다.

5. *isDualAuth*
이중인증 여부를 의미합니다.

구조체

BS2ZoneStatus

```
typedef struct {
    uint32_t id;
    uint8_t status;
    uint8_t disabled;
    uint8_t reserved[6];
} BS2ZoneStatus;
```

1. *id*
구역 식별자입니다.

2. *status*
구역의 현재 상태 값으로 여러 상태값으로 조합될 수 있습니다.

값	설명
0	정상, 경비해제
1	알람 발생
2	Scheduled locked, Lift locked
4	Scheduled unlocked, Lift unlocked
8	경비

3. *disabled*
비활성화 상태인지를 나타내는 flag 값입니다.

4. *reserved*
예약된 공간입니다.

BS2ApbMember

```
typedef struct {
    uint32_t deviceID;
    uint8_t type;
    uint8_t reserved[3];
} BS2ApbMember;
```

1. *deviceID*
장치 식별자입니다.

2. *type*
APB reader 유형입니다.

값	설명
-1	정의되지 않음
0	입구 리더

값	설명
1	출구 리더

3. *reserved*
예약된 공간입니다.

BS2TimedApbMember

```
typedef struct {
    uint32_t deviceID;
    uint8_t reserved[4];
} BS2TimedApbMember;
```

1. *deviceID*
장치 식별자입니다.

2. *reserved*
예약된 공간입니다.

BS2FireSensor

```
typedef struct {
    uint32_t deviceID;
    uint8_t port;
    uint8_t switchType;
    uint8_t duration;
} BS2FireSensor ;
```

1. *deviceID*
장치 식별자입니다.

2. *port*
장치의 입력 포트입니다.

3. *switchType*
입력 신호 유형입니다.

값	설명
0	대기 상태에서 열림 상태로 유지
1	대기 상태에서 닫힘 상태로 유지

4. *duration*
입력 신호를 화재 경보 상태로 판단하기까지 걸리는 시간으로 millisecond 단위입니다.

BS2AntiPassbackZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t type;
    uint8_t numReaders;
    uint8_t numBypassGroups;
    uint8_t disabled;
    uint8_t alarmed;
    uint8_t reserved[3];
    uint32_t resetDuration;
    BS2Action alarm[BS2_MAX_APB_ALARM_ACTION];
    BS2ApbMember readers[BS2_MAX_READERS_PER_APB_ZONE];
    uint8_t reserved2[512];
    uint32_t bypassGroupIDs[BS2_MAX_BYPASS_GROUPS_PER_APB_ZONE];
} BS2AntiPassbackZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

주의
이중 출입 구역의 식별자와 출입문 식별자가 같을 경우 출입문 기반 이중 출입 구역으로 간주하기 때문에 해당 출입문이 제거될 때 구역 정보도 같이 제거될 수 있습니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *type*
안티패스백 구역의 유형입니다.

값	설명
0	Hard APB(규칙 위반시 출입이 불가능하고 로그에 기록됨)
1	Soft APB(규칙 위반시 출입은 허용하고 로그에 기록됨)

4. *numReaders*
APB reader의 총 개수입니다.

5. *numBypassGroups*
APB 규칙을 적용하지 않는 출입 그룹 식별자의 총 개수입니다 .

6. *disabled*
활성화 여부 flag값입니다 .

7. *alarmed*
현재 구역의 경보 상태입니다.

8. *reserved*
예약된 공간입니다.

9. *resetDuration*
APB 위반 상황이 초기화(해제) 될 때까지의 시간을 의미하며, 초 단위로 설정 합니다. 이때 초기화에 소

요되는 시간은 마지막 인증이 성공한 시간을 기준으로 계산됩니다. 만일 이 값이 0으로 설정되었다면 초기화하지 않는 것을 의미하며, BS2_ClearAntiPassbackZoneStatus로 초기화 할 수 있습니다.

10. *alarm*
사용자가 APB 규칙을 위반했을때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

11. *readers*
안티패스백 구역에 속한 장치들을 나열한 리스트로 최대 64개까지 지정할 수 있습니다.

12. *reserved2*
예약된 공간입니다.

13. *bypassGroupIDs*
APB 규칙을 적용하지 않는 출입 그룹 식별자 리스트로 최대 16개까지 지정할 수 있습니다.

BS2TimedAntiPassbackZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t type;
    uint8_t numReaders;
    uint8_t numBypassGroups;
    uint8_t disabled;
    uint8_t alarmed;
    uint8_t reserved[3];
    uint32_t resetDuration;
    BS2Action alarm[BS2_MAX_TIMED_APB_ALARM_ACTION];
    BS2TimedApbMember readers[BS2_MAX_READERS_PER_TIMED_APB_ZONE];
    uint8_t reserved2[320];
    uint32_t bypassGroupIDs[BS2_MAX_BYPASS_GROUPS_PER_TIMED_APB_ZONE];
} BS2TimedAntiPassbackZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *type*
시간 지정 안티패스백 구역의 유형입니다.

값	설명
0	Hard APB(규칙 위반시 출입이 불가능하고 로그에 기록됨)
1	Soft APB(규칙 위반시 출입은 허용하고 로그에 기록됨)

4. *numReaders*
시간 지정 안티패스백 reader의 총 개수입니다.

5. *numBypassGroups*
시간 지정 안티패스백 규칙을 적용하지 않는 출입 그룹 식별자의 총 개수입니다 .

6. *disabled*
활성화 여부 flag값입니다 .

7. *alarmed*
현재 구역의 경보 상태입니다.

8. *reserved*
예약된 공간입니다.

9. *resetDuration*
사용자가 출입 후 동일 장치에 다시 출입하기 위해 대기해야 하는 시간으로 초 단위로 설정할 수 있습니다. 만약, 이 값이 0으로 설정되었다면 초기화하지 않는것을 의미하며, 오직 BioStar 애플리케이션을 통해서만 초기화될 수 있습니다.

10. *alarm*
사용자가 시간 지정 안티패스백 규칙을 위반했을때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

11. *readers*
시간 지정 안티패스백 구역에 속한 장치들을 나열한 리스트로 최대 64개까지 지정할 수 있습니다.

12. *reserved2*
예약된 공간입니다.

13. *bypassGroupIDs*
시간 지정 안티패스백 규칙을 적용하지 않는 출입 그룹 식별자 리스트로 최대 16개까지 지정할 수 있습니다.

BS2FireAlarmZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t numSensors;
    uint8_t numDoors;
    uint8_t disabled;
    uint8_t alarmed;
    uint8_t reserved[8];
    BS2FireSensor sensor[BS2_MAX_FIRE_SENSORS_PER_FIRE_ALARM_ZONE];
    BS2Action alarm[BS2_MAX_FIRE_ALARM_ACTION];
    uint8_t reserved2[32];
    uint32_t doorIDs[BS2_MAX_DOORS_PER_FIRE_ALARM_ZONE];
} BS2FireAlarmZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *numSensors*
화재 경보 센서의 총 개수입니다.

4. *numDoors*
화재 경보 구역에 속한 총 출입문 개수입니다.

5. *alarmed*
현재 구역의 경보 상태입니다.

6. *disabled*
활성화 여부 flag값입니다 .

7. *reserved*
예약된 공간입니다.

8. *sensor*
화재 경보 구역에 속한 화재 경보 센서들을 나열한 리스트로 최대 8개까지 지정할 수 있습니다.

9. *alarm*
화재를 감지했을 때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

10. *reserved2*
예약된 공간입니다.

11. *doorIDs*
화재를 감지했을 때 열어야 할 출입문들을 나열한 리스트로 최대 32개까지 지정할 수 있습니다.

BS2ScheduledLockUnlockZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint32_t lockScheduleID;
    uint32_t unlockScheduleID;
    uint8_t numDoors;
    uint8_t numBypassGroups;
    uint8_t numUnlockGroups;
    uint8_t bidirectionalLock;
    uint8_t disabled;
    uint8_t alarmed;
    uint8_t reserved[6];
    BS2Action alarm[BS2_MAX_SCHEDULED_LOCK_UNLOCK_ALARM_ACTION];
    uint8_t reserved2[32];
    uint32_t doorIDs[BS2_MAX_DOORS_IN_SCHEDULED_LOCK_UNLOCK_ZONE];
    uint32_t
bypassGroupIDs[BS2_MAX_BYPASS_GROUPS_IN_SCHEDULED_LOCK_UNLOCK_ZONE];
    uint32_t
unlockGroupIDs[BS2_MAX_UNLOCK_GROUPS_IN_SCHEDULED_LOCK_UNLOCK_ZONE];
} BS2ScheduledLockUnlockZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *lockScheduleID*
구역에 속한 출입문이 스케줄 잠금 상태로 동작해야 할 출입 일정 식별자입니다.

4. *unlockScheduleID*
구역에 속한 출입문이 스케줄 개방 상태로 동작해야 할 출입 일정 식별자입니다.

5. *numDoors*
구역에 속한 출입문의 총 개수입니다.

6. *numBypassGroups*
스케줄 잠금 상태에서도 출입을 허용하는 출입 그룹의 총 개수입니다.

7. *numUnlockGroups*
스케줄 개방 일정에서 모든 출입문을 열 수 있는 권한을 가진 출입 그룹의 총 개수입니다.

8. *bidirectionalLock*
강제 잠금 상태에서 출입문의 입구 장치와 출구 장치에서 모두 출입을 막아야 하는지 여부입니다.

9. *disabled*
활성화 여부 flag값입니다 .

10. *alarmed*
현재 구역의 경보 상태입니다.

11. *reserved*
예약된 공간입니다.

12. *alarm*
구역 규칙을 위반하였을 때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

13. *reserved2*
예약된 공간입니다.

14. *doorIDs*
구역에 속한 출입문 식별자를 나열한 리스트로 최대 32개까지 설정할 수 있습니다.

15. *bypassGroupIDs*
스케줄 잠금 상태에서도 출입을 허용하는 출입 그룹 식별자를 나열한 리스트로 최대 16개까지 설정할 수 있습니다.

16. *unlockGroupIDs*
스케줄 개방 일정에서 모든 출입문을 열 수 있는 권한을 가진 출입 그룹 식별자를 나열한 리스트로 최대 16개까지 설정할 수 있습니다.

BS2IntrusionAlarmZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t armDelay;
    uint8_t alarmDelay;
    uint8_t disabled;
    uint8_t reserved[1];
    uint8_t numReaders;
    uint8_t numInputs;
    uint8_t numOutputs;
    uint8_t numCards;
    uint8_t numDoors;
    uint8_t numGroups;
    uint8_t reserved2[10];
} BS2IntrusionAlarmZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *armDelay*
경비 수행 지연시간입니다.

4. *alarmDelay*
경비 해제 수행 지연시간입니다.

5. *disabled*
활성화 여부 flag값입니다 .

6. *reserved[1]*
예약된 공간입니다.

7. *numReaders*
경비 경보 구역에 속한 장치들의 총 개수입니다.

8. *numInputs*
경비 경보 구역에 속한 침입감지센서 입력의 총 개수입니다.

9. *numOutputs*
경비 경보 구역에 속한 출력의 총 개수입니다.

10. *numCards*
경비 경보 구역에 속한 카드의 총 개수입니다.

11. *numDoors*
경비 경보 구역에 속한 출입문의 총 개수입니다.

12. *numGroups*
경비 경보 구역에 속한 출입그룹의 총 개수입니다.

13. *reserved*
예약된 공간입니다.

BS2IntrusionAlarmZoneBlob

```
typedef struct {
    BS2IntrusionAlarmZone IntrusionAlarmZone;
    BS2AlarmZoneMember* memberObjs;
    BS2AlarmZoneInput* inputObjs;
    BS2AlarmZoneOutput* outputObjs;
    BS2CSNCard* cardObjs;
    BS2_DOOR_ID* doorIDs;
    BS2_ACCESS_GROUP_ID* groupIDs;
} BS2IntrusionAlarmZoneBlob;
```

1. *IntrusionAlarmZone*
경비 경보 구역의 기본 정보를 정의한 구조체입니다.

2. *memberObjs*
경비 경보 구역에 속한 장치들을 나열한 리스트로 반드시 **IntrusionAlarmZone.numReaders**만큼 존재해야 합니다.

3. *inputObjs*
경비 경보 구역에 속한 입력들을 나열한 리스트로 반드시 **IntrusionAlarmZone.numInputs**만큼 존재해야 합니다.

4. *outputObjs*
경비 경보 구역에 속한 출력들을 나열한 리스트로 반드시 **IntrusionAlarmZone.numOutputs**만큼 존재해야 합니다. 데이터 형식은

5. *cardObjs*
경비 경보 구역에 속한 카드들을 나열한 리스트로 반드시 **IntrusionAlarmZone.numCards**만큼 존재해야 합니다. 데이터 형식은 Smartcard API를 참고하십시오.

6. *doorIDs*
경비 경보 구역에 속한 출입문 식별자를 나열한 리스트로 반드시 **IntrusionAlarmZone.numDoors**만큼 존재해야 합니다.

7. *groupIDs*
경비 경보 구역을 위한 출입 그룹 식별자를 나열한 리스트로 반드시 **IntrusionAlarmZone.numGroups**만큼 존재해야 합니다.

BS2InterlockZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t disabled;
    uint8_t numInputs;
    uint8_t numOutputs;
    uint8_t numDoors;
    uint8_t reserved[8];
} BS2InterlockZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *disabled*
활성화 여부 flag값입니다 .

4. *numInputs*
인터락 구역에 속한 출입문 센서 입력의 총 개수입니다.

5. *numOutputs*
인터락 구역에 속한 출력의 총 개수입니다.

6. *numDoors*
인터락 구역에 속한 출입문의 총 개수입니다.

7. *reserved*
예약된 공간입니다.

BS2InterlockZoneBlob

```
typedef struct {
    BS2InterlockZone InterlockZone;
    BS2InterlockZoneInput* inputObjs;
    BS2InterlockZoneOutput* outputObjs;
    BS2_DOOR_ID* doorIDs;
} BS2InterlockZoneBlob;
```

1. *InterlockZone*
인터락 구역의 기본 정보를 정의한 구조체입니다.

2. *inputObjs*
인터락 구역에 속한 입력들을 나열한 리스트로 반드시 **InterlockZone.numInputs**만큼 존재해야 합니다.

3. *outputObjs*
인터락 구역에 속한 출력들을 나열한 리스트로 반드시 **InterlockZone.numOutputs**만큼 존재해야 합니다.

4. *doorIDs*
인터락 구역에 속한 출입문 식별자를 나열한 리스트로 반드시 **InterlockZone.numDoors**만큼 존재해야 합니다.

BS2DeviceZoneEntranceLimitMaster

```
typedef struct {
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t type;
    uint8_t reserved1[3];
    uint32_t entryLimitInterval_s;
    uint8_t numEntranceLimit;
    uint8_t numReaders;
    uint8_t numAlarm;
    uint8_t numBypassGroups;
    uint8_t maxEntry[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    uint32_t periodStart_s[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    uint32_t periodEnd_s[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    BS2DeviceZoneEntranceLimitMemberInfo
readers[BS2_MAX_READERS_PER_DEVICE_ZONE_ENTRANCE_LIMIT];
    BS2Action alarm[BS2_MAX_DEVICE_ZONE_ENTRANCE_LIMIT_ALARM_ACTION];
    BS2_ACCESS_GROUP_ID
bypassGroupIDs[BS2_MAX_BYPASS_GROUPS_PER_DEVICE_ZONE_ENTRANCE_LIMIT];
    uint8_t reserved3[8 * 4];
} BS2DeviceZoneEntranceLimitMaster;
```

1. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

2. *type*
인증 제한 구역의 유형입니다

값	설명
1	Soft EntranceLimit(규칙 위반시 출입이 불가능하고 로그에 기록됨)
2	Hard EntranceLimit(규칙 위반시 출입은 허용하고 로그에 기록됨)

3. *reserved1[3]*
예약된 공간입니다.

4. *entryLimitInterval_s*
인증 출입간의 간격입니다.

5. *numEntranceLimit*
인증 제한 총 개수입니다.

6. *numReaders*
인증 제한 구역 reader의 총 개수입니다.

7. *numAlarm*
인증 제한 구역 경보의 총 개수입니다.

8. *numBypassGroups*
인증 제한 구역 규칙을 적용하지 않는 출입 그룹 식별자의 총 개수입니다.

9. *maxEntry*
출입 지정 최대 횟수입니다.

10. *periodStart_s*
출입 가능한 시작 시간입니다. (초단위)

11. *periodEnd_s*
출입 가능한 종료 시간입니다. (초단위)

12. *readers*
인증 제한 구역에 속한 장치들을 나열한 리스트로 최대 64개까지 지정할 수 있습니다.

13. *alarm*
구역 규칙을 위반하였을 때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

14. *bypassGroupIDs*
인증 제한 규칙을 적용하지 않는 출입 그룹 식별자 리스트로 최대 16개까지 지정할 수 있습니다.

15. *reserved3*
예약된 공간입니다.

BS2DeviceZoneEntranceLimitMember

```
typedef struct {
    uint16_t masterPort;
    BS2_DEVICE_ZONE_ENTRANCE_LIMIT_DISCONNECTED_ACTION_TYPE
actionInDisconnect;
    uint8_t reserved1[1];
    BS2_IPV4_ADDR masterIP;
} BS2DeviceZoneEntranceLimitMember;
```

1. *masterPort*
master 장치의 port입니다.

2. *actionInDisconnect*
Disconnect일때의 액션입니다 .

값	설명
1	Soft EntranceLimit Disconnected action(규칙 위반시 출입이 불가능하고 로그에 기록됨)
2	Hard EntranceLimit Disconnected action(규칙 위반시 출입은 허용하고 로그에 기록됨)

3. *reserved1[3]*
예약된 공간입니다.

4. *masterIP*
master 장치의 IP입니다.

BS2DeviceZoneFireAlarmMaster

```
typedef struct {
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint8_t numReaders;
    uint8_t numAlarm;
    uint8_t reserved1[2];
    BS2DeviceZoneFireAlarmMemberInfo
readers[BS2_MAX_READERS_PER_DEVICE_ZONE_FIRE_ALARM];
    BS2Action alarm[BS2_MAX_DEVICE_ZONE_FIRE_ALARM_ALARM_ACTION];
    uint8_t reserved2[8 * 40];
} BS2DeviceZoneFireAlarmMaster;
```

1. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

2. *numReaders*
화재 경보 구역에 속한 장치의 총 개수입니다.

3. *reserved1*
예약된 공간입니다.

4. *readers*
화재 경보 구역에 속한 장치입니다.

5. *alarm*
화재를 감지했을 때 동작할 경보로 최대 5개까지 설정할 수 있습니다

6. *reserved2*
예약된 공간입니다.

BS2DeviceZoneFireAlarmMember

```
typedef struct {
    BS2_PORT masterPort;
    uint8_t reserved1[2];
    BS2_IPV4_ADDR masterIP;
    uint8_t numSensors;
    uint8_t numDoors;
    uint8_t reserved2[2];
    BS2DeviceZoneFireSensor
sensor[BS2_MAX_FIRE_SENSORS_PER_DEVICE_ZONE_FIRE_ALARM_MEMBER];
    union {
        BS2_DOOR_ID
doorIDs[BS2_MAX_DOORS_PER_DEVICE_ZONE_FIRE_ALARM_MEMBER];
        BS2_LIFT_ID
liftIDs[BS2_MAX_DOORS_PER_DEVICE_ZONE_FIRE_ALARM_MEMBER];
    };

} BS2DeviceZoneFireAlarmMember;
```

1. *masterPort*
master 장치의 port입니다.

2. *reserved1*
예약된 공간입니다.

3. *masterIP*
master 장치의 IP입니다.

4. *numSensors*
화재 경보 센서의 총 개수입니다.

5. *numDoors*
화재 경보 구역에 속한 총 출입문 개수입니다.

6. *reserved2*
예약된 공간입니다.

7. *sensor*
화재 경보 구역에 속한 화재 경보 센서들을 나열한 리스트로 최대 8개까지 지정할 수 있습니다.

8. *doorIDs*
화재를 감지했을 때 열어야 할 출입문들을 나열한 리스트로 최대 8개까지 지정할 수 있습니다.

9. *liftIDs*
화재를 감지했을 때 열어야 할 Lift들을 나열한 리스트로 최대 8개까지 지정할 수 있습니다.

BS2DeviceZoneFireAlarmMemberInfo

```
typedef struct {
    uint32_t readerID;
} BS2DeviceZoneFireAlarmMemberInfo;
```

1. *readerID*
화재 경보에 속한 장치입니다.

BS2DeviceZoneFireSensor

```
typedef struct {
    uint32_t deviceID;
    uint8_t port;
    uint8_t switchType;
    uint16_t duration;
} BS2DeviceZoneFireSensor;
```

1. *deviceID*

장치 식별자입니다.

2. *port*
장치의 입력 포트입니다.

3. *switchType*
입력 신호 유형입니다.

값	설명
0	대기 상태에서 열림 상태로 유지
1	대기 상태에서 닫힘 상태로 유지

4. *duration*
입력 신호를 화재 경보 상태로 판단하기까지 걸리는 시간으로 millisecond 단위입니다.

BS2DeviceZone

```
typedef struct {
    uint32_t zoneID;
    uint8_t zoneType;
    uint8_t nodeType;
    uint8_t enable;
    uint8_t reserved[1];
    union {
        BS2DeviceZoneEntranceLimitMaster entranceLimitMaster;
        BS2DeviceZoneEntranceLimitMember entranceLimitMember;
        BS2DeviceZoneFireAlarmMaster fireAlarmMaster;
        BS2DeviceZoneFireAlarmMember fireAlarmMember;
    };
} BS2DeviceZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *zoneType*
경비 수행 지연시간입니다.

3. *nodeType*
경비 해제 수행 지연시간입니다.

4. *enable*
활성화 여부 flag값입니다 .

5. *reserved[1]*
예약된 공간입니다.

BS2DeviceZoneAGEntranceLimit

```
typedef struct {
    uint32_t zoneID;
    uint16_t numAGEntranceLimit;
    uint16_t reserved1;
    uint32_t periodStart_s[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    uint32_t periodEnd_s[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    uint16_t numEntry[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE];
    uint16_t
maxEntry[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE][BS2_MAX_ACCESS_GROUP_ENTRANCE_LIMIT_PER_ENTRACE_LIMIT];
    uint32_t
accessGroupID[BS2_MAX_ENTRANCE_LIMIT_PER_ZONE][BS2_MAX_ACCESS_GROUP_ENTRANCE_LIMIT_PER_ENTRACE_LIMIT];
} BS2DeviceZoneAGEntranceLimit;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *numAGEntranceLimit*
출입그룹 인증 제한 총 개수입니다.

3. *reserved1*
예약된 공간입니다.

4. *periodStart_s*
출입 가능한 시작 시간입니다.

5. *periodEnd_s*
출입 가능한 종료 시간입니다.

6. *numEntry*
출입 총 개수입니다.

7. *maxEntry*
출입 지정 최대 횟수입니다.

8. *accessGroupID*
출입 그룹 식별자 리스트로 최대 16개까지 지정할 수 있습니다

BS2DeviceZoneMasterConfig

```
typedef struct
{
    bool enable;
    uint8_t reserved1[1];
    uint16_t listenPort;
    uint8_t reserved[4];
} BS2DeviceZoneMasterConfig;
```

1. *enable*
활성화 여부 flag값입니다.

2. *reserved1*
예약된 공간입니다.

3. *listenPort*
Slave 장치들로부터 연결되어질 TCP/IP port 정보입니다.

4. *reserved*
예약된 공간입니다.

BS2LiftFloors

```
typedef struct {
    uint32_t liftID;
    uint16_t numFloors;
    uint16_t reserved;
    uint8_t floorIndices[256];
} BS2LiftFloors;
```

1. *liftID*
lift 식별자로 항상 1이상의 값을 가져야 합니다.

2. *numFloors*
floorIndices의 개수입니다.

3. *reserved*
예약된 공간입니다.

4. *floorIndices*
BS2Lift의 BS2LiftFloor 255개 배열에서 원하는 floor 정보의 인덱스 값을 이곳에서 순서대로 나열합니다. 예를들어, 1, 3, 5, 9번째 BS2LiftFloor 정보로 scheduled lock/unlock 구역의 멤버 floor를 구성하고 싶다면,
floorIndices[0]부터 순서대로 1, 3, 5, 9로 지정되어야 합니다.

BS2LiftLockUnlockZone

```
typedef struct {
    uint32_t zoneID;
    char name[BS2_MAX_ZONE_NAME_LEN];
    uint32_t unlockScheduleID;
    uint32_t lockScheduleID;

    uint8_t numLifts;
    uint8_t numBypassGroups;
    uint8_t numUnlockGroups;
    uint8_t unused;

    uint8_t disabled;
    uint8_t alarmed;
    uint8_t reserved[6];

    BS2Action alarm[BS2_MAX_LIFT_LOCK_UNLOCK_ALARM_ACTION];

    uint8_t reserved2[32];

    BS2LiftFloors lifts[BS2_MAX_LIFTS_IN_LIFT_LOCK_UNLOCK_ZONE];
    uint32_t bypassGroupIDs[BS2_MAX_BYPASS_GROUPS_IN_LIFT_LOCK_UNLOCK_ZONE];
    uint32_t unlockGroupIDs[BS2_MAX_UNLOCK_GROUPS_IN_LIFT_LOCK_UNLOCK_ZONE];
} BS2LiftLockUnlockZone;
```

1. *zoneID*
구역 식별자로 항상 1이상의 값을 가져야 합니다.

2. *name*
BioStar 애플리케이션에서 보여줄 구역 이름입니다.

3. *unlockScheduleID*
구역으로 지정한 lift floor들이 스케줄 개방 상태로 동작해야 할 스케줄정보 식별자입니다.

4. *lockScheduleID*
구역으로 지정한 lift floor들이 스케줄 잠금 상태로 동작해야 할 스케줄정보 식별자입니다.

5. *numLifts*
lifts로 지정한 정보의 개수입니다.

6. *numBypassGroups*
lift스케줄 잠금 상태에서도 출입을 허용하는 출입 그룹의 총 개수입니다.

7. *numUnlockGroups*
lift스케줄 개방 일정에서 모든 출입문을 열 수 있는 권한을 가진 출입 그룹의 총 개수입니다.

8. *unused*
사용하지 않는 필드입니다.

9. *disabled*
활성화 여부 flag값입니다 .

10. *alarmed*
현재 구역의 경보 상태입니다.

11. *reserved*
예약된 공간입니다.

12. *alarm*
구역 규칙을 위반하였을 때 동작할 경보로 최대 5개까지 설정할 수 있습니다.

13. *reserved2*
예약된 공간입니다.

14. *lifts*
lift의 floor 정보입니다.

15. *bypassGroupIDs*

lift스케줄 잠금 상태에서도 출입을 허용하는 출입 그룹 식별자를 나열한 리스트로 최대 16개까지 설정할 수 있습니다.

16. *unlockGroupIDs*
lift스케줄 개방 일정에서 모든 출입문을 열 수 있는 권한을 가진 출입 그룹 식별자를 나열한 리스트로 최대 16개까지 설정할 수 있습니다.

From:
http://kb.supremainc.com/bs2sdk/ - **BioStar Device SDK**

Permanent link:
http://kb.supremainc.com/bs2sdk/doku.php?id=ko:zone_control_api

Last update: **2023/08/31 09:39**