목차

Door Control API 1
구조체 1
BS2DoorRelay 1
BS2DoorSensor 1
BS2ExitButton 2
BS2DoorStatus 2
BS2Door 4

Door Control API

출입문의 내외부에 어떤 장치가 설치되어 있는지, 장치가 출입문을 어떻게 제어할것인지, 이중 인증 방지 구역을 어떻게 적용할 것인지 설정할 수 있습니다..

- BS2_GetDoor: 주어진 출입문을 가져옵니다.
- BS2_GetAllDoor: 모든 출입문을 가져옵니다.
- BS2_GetDoorStatus: 주어진 출입문의 상태 정보를 가져옵니다.
- BS2_GetAllDoorStatus: 모든 출입문의 상태 정보를 가져옵니다.
- BS2_SetDoor: 출입문을 설정합니다.
- BS2_SetDoorAlarm: 출입문의 경보 상태를 갱신합니다.
- BS2_RemoveDoor: 주어진 출입문를 제거합니다.
- BS2_RemoveAllDoor: 모든 출입문를 제거합니다.
- BS2_ReleaseDoor: 출입문의 우선 순위를 제거합니다.
- BS2_LockDoor: 출입문이 닫힐 때의 우선 순위를 설정하며, 열릴 때의 우선 순위보다 높아야 문이 닫힙니다.
- BS2_UnlockDoor: 출입문이 열릴 때의 우선 순위를 설정하며, 닫힐 때의 우선 순위보다 높아야 문이 열립니다.

구조체

BS2DoorRelay

```
typedef struct {
    uint32_t deviceID;
    uint8_t port;
    uint8_t reserved[3];
} BS2DoorRelay;
```

1. *deviceID*
장치 식별자입니다.

2. *port*
릴레이 port number입니다.

3. *reserved*
예약된 공간입니다.

BS2DoorSensor

```
typedef struct {
    uint32_t deviceID;
    uint8_t port;
    uint8_t switchType;
    uint8_t reserved[2];
} BS2DoorSensor ;
```

1. *deviceID*
장치 식별자입니다.

2. *port*
입력 port number입니다.

3. *switchType*
입력 신호 유형입니다.

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

4. *reserved*
예약된 공간입니다.

BS2ExitButton

```
typedef struct {
    uint32_t deviceID;
    uint8_t port;
    uint8_t switchType;
    uint8_t reserved[2];
} BS2ExitButton ;
```

1. *deviceID*
장치 식별자입니다.

2. *port*
입력 port number입니다.

3. *switchType*
입력 신호 유형입니다.

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

4. *reserved*
예약된 공간입니다.

BS2DoorStatus

```
typedef struct {
    uint32_t id;
    uint8_t opened;
    uint8_t unlocked;
    uint8_t heldOpened;
```

```
    uint8_t unlockFlags;
    uint8_t lockFlags;
    uint8_t alarmFlags;
    uint8_t reserved[2];
    uint32_t lastOpenTime;
} BS2DoorStatus;
```

1. *id*
출입문 식별자입니다.

2. *opened*
출입문이 열려있는지 나타내는 flag 입니다.

3. *unlocked*
출입문이 개방되어있는지 나타내는 flag 입니다.

4. *heldOpened*
출입문이 장시간 열림 상태인지 나타내는 flag 입니다.

5. *unlockFlags*
출입문이 열릴 때의 우선 순위로 닫힐 때의 우선 순위보다 낮으면 수행되지 않습니다. 예를 들어, 관리자 권한으로 출입문을 닫은 상태라면(OPERATOR), 모든 사용자의 출입이 허용되지 않습니다. 유의해야 할 사항으로 unlockFlags와 lockFlags는 기본 우선 순위(NONE)를 제외하고는 동일 우선 순위를 가질 수 없습니다.

값	설명	우선순위
0	기본	보통
1	일정	약간 높음
4	관리자	높음
2	긴급	매우 높음

6. *lockFlags*
출입문이 닫힐 때의 우선 순위로 열릴 때의 우선 순위보다 낮으면 수행되지 않습니다.

값	설명	우선순위
0	기본	보통
1	일정	약간 높음
4	관리자	높음
2	긴급	매우 높음

7. *alarmFlags*
현재 출입문의 경보 상태 정보입니다.

값	설명
0	경보 없음
1	강제 문열림
4	장시간 문열림
2	APB 위반

8. *reserved*

예약된 공간입니다.

9. *lastOpenTime*
출입문이 마지막으로 열렸던 시간입니다.

BS2Door

```
typedef struct {
    uint32_t doorID;
    char name[BS2_MAX_DOOR_NAME_LEN];
    uint32_t entryDeviceID;
    uint32_t exitDeviceID;
    BS2DoorRelay relay;
    BS2DoorSensor sensor;
    BS2ExitButton button;
    uint32_t autoLockTimeout;
    uint32_t heldOpenTimeout;
    uint8_t instantLock;
    uint8_t unlockFlags;
    uint8_t lockFlags;
    uint8_t unconditionalLock;
    BS2Action forcedOpenAlarm[BS2_MAX_FORCED_OPEN_ALARM_ACTION];
    BS2Action heldOpenAlarm[BS2_MAX_HELD_OPEN_ALARM_ACTION];
    uint32_t dualAuthScheduleID;
    uint8_t dualAuthDevice;
    uint8_t dualAuthApprovalType;
    uint32_t dualAuthTimeout;
    uint8_t numDualAuthApprovalGroups;
    uint8_t reserved2[1];
    uint32_t dualAuthApprovalGroupID[BS2_MAX_DUAL_AUTH_APPROVAL_GROUP];
    BS2AntiPassbackZone apbZone;
} BS2Door;
```

1. *doorID*
출입문 식별자입니다.

2. *name*
BioStar 애플리케이션에서 보여질 출입문 이름입니다.

3. *entryDeviceID*
입구 장치 식별자입니다.

4. *exitDeviceID*
출구 장치 식별자입니다.

5. *relay*
출입문 릴레이입니다.

6. *sensor*
출입문 열림/닫힘 상태를 감지하는 센서입니다.

7. *button*
Exit 버튼입니다.

8. *autoLockTimeout*
출입문이 열린 후 자동으로 닫힐때까지 걸리는 시간으로 초 단위입니다.

9. *heldOpenTimeout*
장시간 출입문 열림 상태를 판별하는 시간으로 초 단위입니다.

10. *instantLock*
출입문 센서가 닫힌 상태로 감지할 때 출입문을 즉시 닫을지 결정하는 flag입니다.

11. *unlockFlags*
출입문이 열릴 때의 우선 순위로 닫힐 때의 우선 순위보다 낮으면 수행되지 않습니다. 예를 들어, 관리자 권한으로 출입문을 닫은 상태라면(OPERATOR), 모든 사용자의 출입이 허용되지 않습니다. 유의해야 할 사항으로 unlockFlags와 lockFlags는 기본 우선 순위(NONE)를 제외하고는 동일 우선 순위를 가질 수 없습니다.

값	설명	우선순위
0	기본	보통
1	일정	약간 높음
4	관리자	높음
2	긴급	매우 높음

12. *lockFlags*
출입문이 닫힐 때의 우선 순위로 열릴 때의 우선 순위보다 낮으면 수행되지 않습니다.

값	설명	우선순위
0	기본	보통
1	일정	약간 높음
4	관리자	높음
2	긴급	매우 높음

13. *unconditionalLock*
autoLock timeout 이후에 door lock 결정하는 flag입니다.

값	설명
0	autoLock timeout 이후에 door close할 경우에만 door lock을 함. (door open시 실제로 door가 close될때까지 기다림)
1	autoLock timeout 이후에 door open/close에 상관없이 door lock을 함

14. *forcedOpenAlarm*
강제로 문이 열릴 때 수행되는 경보로 최대 5개까지 지정할 수 있습니다.

15. *heldOpenAlarm*
장시간 문이 열린 상태로 유지될 때 수행되는 경보로 최대 5개까지 지정할 수 있습니다.

16. *dualAuthScheduleID*
이중 인증이 필요한 타임 스케줄로 이중 인증이 필요 없다면 0을, 항상 필요하면 1을, 아니면 적절한 타

임 스케줄 식별자를 지정할 수 있습니다.

17. *dualAuthDevice*
어떤 장치가 이중 인증을 수행해야 하는지를 나타내는 flag입니다.

값	설명
0	없음
1	입구 장치만 이중 인증
2	출구 장치만 이중 인증
3	입출구 장치 모두 이중 인증

18. *dualAuthApprovalType*
출입을 허용할 때 사용자의 출입 그룹이 이중 인증 출입 권한을 가지는 그룹에 속한지 검사하는 flag입니다.

값	설명
0	없음
1	마지막 사용자의 권한 확인

19. *dualAuthTimeout*
1번째 사용자의 인증이 끝나고 2번째 사용자의 인증을 시도하기까지 대기하는 시간으로 초 단위입니다.

20. *numDualAuthApprovalGroups*
이중 출입 권한을 가지는 그룹의 총 개수입니다.

21. *reserved2*
예약된 공간입니다.

22. *dualAuthApprovalGroupID*
이중 출입 권한을 가지는 그룹을 나열한 리스트로 최대 16까지 설정할 수 있습니다.

23. *apbZone*
이 출입문을 출입문 기반 이중 출입 방지 구역으로 설정합니다. 주의 사항으로 출입문 식별자와 구역 식별자는 동일한 값을 가져야 합니다. 자세한 정보는 Zone Control API를 참고하십시오.