

Door Control API 1

..... 1

BS2DoorRelay 1

BS2DoorSensor 1

BS2ExitButton 2

BS2DoorStatus 2

BS2Door 4

Door Control API

가 , 가 ,

- [BS2_GetDoor:](#) 가 .
- [BS2_GetAllDoor:](#) 가 .
- [BS2_GetDoorStatus:](#) 가 .
- [BS2_GetAllDoorStatus:](#) 가 .
- [BS2_SetDoor:](#) .
- [BS2_SetDoorAlarm:](#) .
- [BS2_RemoveDoor:](#) .
- [BS2_RemoveAllDoor:](#) .
- [BS2_ReleaseDoor:](#) .
- [BS2_LockDoor:](#) ,
- [BS2_UnlockDoor:](#) ,

BS2DoorRelay

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

1. *deviceID*

2. *port*

port number

3. *reserved*

BS2DoorSensor

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

1. *deviceID*

2. *port*

port number

3. *switchType*

0	
1	

4. *reserved*

BS2ExitButton

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

1. *deviceID*

2. *port*

port number

3. *switchType*

0	
1	

4. *reserved*

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*

2. *opened*

flag

3. *unlocked*

flag

4. *heldOpened*

flag

5. *unlockFlags*

(OPERATOR),

unlockFlags

lockFlags

(NONE)

가

0		
1		
4		
2		

6. *lockFlags*

0		
1		
4		
2		

7. *alarmFlags*

0	
1	
4	
2	APB

8. *reserved*

9. *lastOpenTime***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*

. 1 65535

2. *name*

BioStar

. 1

가

3. *entryDeviceID*4. *exitDeviceID*5. *relay*6. *sensor*

/

7. *button*

Exit

8. *autoLockTimeout*

9. *heldOpenTimeout*

10. *instantLock*

가

flag

11. *unlockFlags*

(OPERATOR),

unlockFlags

lockFlags

(NONE)

가

0		
1		
4		
2		

12. *lockFlags*

0		
1		
4		
2		

13. *unconditionalLock*

autoLock timeout

door lock

flag

0	autoLock timeout door close door lock . (door open door가 close)
1	autoLock timeout door open/close door lock

14. *forcedOpenAlarm*

5

15. *heldOpenAlarm*

5

16. *dualAuthScheduleID*

0 ,

1 ,

17. *dualAuthDevice*

가

flag

0	
1	
2	
3	

18. *dualAuthApprovalType*

가

flag

0	
1	

19. *dualAuthTimeout*

1 2

20. *numDualAuthApprovalGroups*

가

21. *reserved2*

22. *dualAuthApprovalGroupID*

가

16

23. *apbZone*

가

[Zone Control API](#)

From:

<http://kb.supremainc.com/bs2sdk/> - **BioStar 2 Device SDK**

Permanent link:

http://kb.supremainc.com/bs2sdk/doku.php?id=ko:door_control_api&rev=1515638262

Last update: **2018/01/11 11:37**