Door Control API

Door Control API

,   ,
..

- [BS2_GetDoor](): .
- [BS2_GetAllDoor](): .
- [BS2_GetDoorStatus](): .
- [BS2_GetAllDoorStatus](): .
- [BS2_SetDoor](): .
- [BS2_SetDoorAlarm](): .
- [BS2_RemoveDoor](): .
- [BS2_RemoveAllDoor](): .
- [BS2_ReleaseDoor](): .
- [BS2_LockDoor](): ,
  .
- [BS2_UnlockDoor](): ,
  .
- [BS2_TimedLockDoor](): [+ 2.9.12] .
- [BS2_TimedUnlockDoor](): [+ 2.9.12] .

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 apbUseDoorSensor;
```

```
    uint8_t reserved[1];
} BS2DoorSensor;
```

1. *deviceID*
   .

2. *port*
   port number   .

3. *switchType*
      .

0	
1	

4. *apbUseDoorSensor*
APB   door sensor   .
5. *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)
　.

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;
    uint16_t extendedAutoLockTimeout;
    uint32_t dualAuthTimeout;
    uint8_t numDualAuthApprovalGroups;
    uint8_t unused[2];
    uint8_t reserved2[1];
    uint32_t dualAuthApprovalGroupID[BS2_MAX_DUAL_AUTH_APPROVAL_GROUP];
    BS2AntiPassbackZone apbZone;
} BS2Door;
```

1. *doorID*

. 1 65535 .

2. *name*
BioStar . 1 .

3. *entryDeviceID*

.

4. *exitDeviceID*

.

5. *relay*

.

6. *sensor*

/ .

7. *button*
Exit .

8. *autoLockTimeout*

.

9. *heldOpenTimeout*

.

10. *instantLock*

flag .

11. *unlockFlags*

. ,
(OPERATOR), .
unlockFlags lockFlags (NONE)
.

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 door close)

1	autoLock timeout door open/close door lock

14. *forcedOpenAlarm*

5 .

15. *heldOpenAlarm*

5 .

16. *dualAuthScheduleID*

0 , 1 ,

.

17. *dualAuthDevice*

flag .

0	
1	
2	
3	

18. *dualAuthApprovalType*

flag

.

0	
1	

19. *extendedAutoLockTimeout*
[+ 2.9.12] . 10 .

20. *dualAuthTimeout*

, . (:)

21. *numDualAuthApprovalGroups*

.

22. *unused*

.

23. *reserved2*

.

24. *dualAuthApprovalGroupID*

16 .

25. *apbZone*

.

. Zone Control API .

From:
<https://kb.supremainc.com/kbtest/> - **BioStar Device SDK**

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=ko:door_control_api

Last update: **2026/01/28 16:55**