Table of Contents

Version 2.3

Date

2016-08-03

New Features

- Supports Multi Wiegand Format
- Supports BioEntry W2

Added & Changed Features

1. Multi-wiegand feature has been added. You will now be able to use multiple types of formats with one device. If the device supports different card types, you will also be able to use multiple types of formats with various types of cards. Maximum 15 types of formats are supported.

- API BS2_GetWiegandMultiConfig has been added
- API BS2_SetWiegandMultiConfig has been added
- Structure BS2WiegandMultiConfig has been added

2. Field **wiegandMultiSupported** has been added to the BS2SimpleDeviceInfo structure
The BS2SimpleDeviceInfo structure is used to show information of the device following by which feature the device supports or not. Since the multi-wiegand feature has been implemented to the devices, this field has been added to show whether the device supports the multi-wiegand feature.

3. Mask **Value 0x0D**(BioEntry W2) has been added for the type field of BS2SimpleDeviceInfo structure
The BS2SimpleDeviceInfo structure is used to give simple information about the device. Since the newly released BioEntry W2 is now supported from the BioStar 2 Device SDK, the mask for this product has been added.

4. Parameter **IsAcceptableUserID** has been added to the BS2_GetUserList API
This callback function is irrelevant to BioStar 2 Device SDK developers. This is used by the BioStar v1.91 which is the latest BioStar 1 integrated with the BioStar 2 devices such as BioStation 2, BioStation A2, BioStation L2. The maximum user ID of BioStar 1 is 4294967295, but the V2 devices' maximum user ID can go up to 32 digits when creating a user directly from the device. To avoid uploading users having a user ID that is beyond the limit of BioStar 1, this callback function had to be added. BioStar 2 Device SDK developers just can put null for this parameter.

BEFORE

```
#include "BS_API.h"

int BS2_GetUserList(void* context, uint32_t deviceId, char** uidsObjs,
uint32_t* numUid);
```

AFTER

```
#include "BS_API.h"

int BS2_GetUserList(void* context, uint32_t deviceId, char** uidsObjs,
uint32_t* numUid, IsAcceptableUserID ptrIsAcceptableUserID);
```

EXAMPLE

```
BS2_GetUserList(context, deviceId, uidsObjs, numUid, null);
```

5. API BS2_GetLastFingerprintImage has been added

An API that gets the actual image of the fingerprint has been added. This function has been implemented to provide a feature that has been existed in the BioStar 2 SW. This will allow you to monitor the actual status of the fingerprint by showing the image of the fingerprint instead of the fingerprint template. However, this image will not be used in any kind of matching.

6. Parameter **templateFormat** has been added to the BS2_ScanFingerprint API

Suprema's fingerprint devices supports 3 types of fingerprint formats such as Suprema Template, ISO 19794-2, and ANSI 384.

Value	Description
0	Suprema Template
1	ISO19794-2
2	ANSI-378

BEFORE

```
#include "BS_API.h"

int BS2_ScanFingerprint(void* context, uint32_t deviceId, BS2Fingerprint*
finger, uint32_t templateIndex, uint32_t quality, OnReadyToScan
ptrReadyToScan);
```

AFTER

```
#include "BS_API.h"

int BS2_ScanFingerprint(void* context, uint32_t deviceId, BS2Fingerprint*
finger, uint32_t templateIndex, uint32_t quality, uint8_t templateFormat,
OnReadyToScan ptrReadyToScan);
```

7. Field **wiegandMultiConfig** has been added to BS2Configs structure

All structures related to configurations has been managed from the BS2Configs structure. Since a new structure for multi wiegand has been added, this structure was added to the BS2Configs structure.

8. Field **wiegandInputMask** has been added to the BS2WiegandConfig structure
Since multi-wiegand is now supported, there needs to be a bitmask field to set which format indexes will be used. This field is used to select the formats that will be understood by the master device when a card ID is sent through wiegand by a 3rd party wiegand reader or a slave device that is connected via RS-485. This can be used when the device is set to receive wiegand signals. For example, if you want to use format index 0, 2, 6, you need to set the bitmask to 0000000010001010. The 0th bit is reserved for the backward compatibility.

9. Field **wiegandCardMask** has been added to the BS2WiegandConfig structure
Since multi-wiegand is now supported, there needs to be a bitmask field to set which format indexes will be used. This field is used to select the formats that will be understood by the device when the card is directly placed to the device. This can be used when the device is set to send out wiegand signals. For example, if you want to use format index 0, 1, 2, 3, you need to set the value to 0000000000011110. The 0th bit is reserved for the backward compatibility.

10. Field **wiegandCSNIndex** has been added to the BS2WiegandConfig structure
When using a card that uses only the CSN of the card, you can still set the device to send out a wiegand signal in a specific format. This can be used when the device can only read Mifare cards or EM cards, but still needs to send out wiegand signals to another device in a certain format. For example, if you want to use format index 0, you need to set the value as 0.

11. Field **reserved** from the BS2WiegandConfig structure has been changed to **uint8_t reserved[27]** from **uint8_t reserved[32]**
There were 3 additional fields that have been added to the BS2WiegandConfig structure, which are mentioned right above. Because of this, there were 5 bytes used from the reserved space, and the reserved space now has been reduced to 27 bytes.

12. Field **alarmFlags** from the BS2Door sturcture has been changed to **uint8_t unconditionalLock** from **uint8_t alarmFlags**
The **alarmFlags** is not used in the BS2Door structure as now the **alarmFlags** can be displayed from the BS2DoorStatus structure. Instead, the **unconditionalLock** field has been added for the new feature and is using the memory of the **alarmFlags**. The **unconditionalLock** field has been added to control automatic slide doors. When using a traditional dead-bolt lock, the lock shouldn't be locked after the configured time has passed unless the door is closed. However, when using an automatic slide door, there are needs for the door to be closed even when there is still an input on the door sensor. You can change this field to true to close the door regardless to the door sensor input.