목차

Face API 1
FaceEx API 1
구조체 1
BS2Face 1
BS2AuthGroup 2
BS2TemplateEx 2
BS2FaceEx 3

Face API

사용자의 얼굴을 인식하여 템플릿 데이터를 추출하고 매치하는 함수를 제공하는 API입니다.

- BS2_ScanFace: FaceStation2 FaceLite 장치에서 얼굴을 스캔하고 템플릿 데이터와 이미지 데이터를 추출합니다.
- BS2_GetAuthGroup: 주어진 인증 그룹을 가져옵니다.
- BS2_GetAllAuthGroup: 모든 인증 그룹을 가져옵니다.
- BS2_SetAuthGroup: 인증 그룹을 설정합니다.
- BS2_RemoveAuthGroup: 주어진 인증 그룹을 제거합니다.
- BS2_RemoveAllAuthGroup: 모든 인증 그룹을 제거합니다.

FaceEx API

- BS2_ScanFaceEx: FaceStation F2 장치에서 얼굴을 스캔하고 이미지 및 템플릿을 추출합니다. [+ 2.7.1]
- BS2_ExtractTemplateFaceEx: FaceStation F2 얼굴이 포함된 이미지를 가지고 template을 추출합니다. [+ 2.7.1]
- BS2_GetNormalizedImageFaceEx: FaceStation F2 WARP 되어지지 않은(가공되어지지 않은) 얼굴 이미지로, WARP 이미지를 만듭니다. [+ 2.8]

구조체

BS2Face

```
typedef struct {
    uint8_t faceIndex;
    uint8_t numOfTemplate;
    uint8_t flag;
    uint8_t reserved;

    uint16_t imageLen;
    uint8_t reserved2[2];

    uint8_t imageData[BS2_FACE_IMAGE_SIZE];
    uint8_t templateData[BS2_TEMPLATE_PER_FACE][BS2_FACE_TEMPLATE_LENGTH];
} BS2Face;
```

1. *faceIndex*
얼굴 인덱스입니다.

2. *numOfTemplate*
얼굴 템플릿 개수입니다.

3. *flag*
FaceStation2 V1.4.0, FaceLite V1.2.0 기준으로 장치내부에서 사용되는 값입니다.

장치에서 수신한 flag 값을 사용하면 되지만, 알 수 없다면 0으로 지정하면 됩니다.

4. *reserved*
예약된 공간입니다.

5. *imageLen*
얼굴 이미지 데이터 크기입니다.

6. *reserved2*
예약된 공간입니다.

4. *imageData*
얼굴 이미지 데이터입니다.

5. *data*
얼굴 템플릿 데이터입니다.

BS2AuthGroup

```
typedef struct {
    BS2_AUTH_GROUP_ID        id;
    char                name[BS2_MAX_AUTH_GROUP_NAME_LEN];
    uint8_t                reserved[32];
} BS2AuthGroup;
```

1. *id*
인증 그룹 식별자입니다.

2. *name*
BioStar 애플리케이션에서 보여줄 인증 그룹 이름입니다.

3. *reserved*
예약된 공간입니다.

BS2TemplateEx

```
typedef struct {
    uint8_t       data[552];
    uint8_t       isIR;
    uint8_t       reserved[3];
} BS2TemplateEx;
```

FaceStation F2

1. *data*
IR 또는 visual 이미지 탬플릿 데이터 입니다.

2. *isIR*
IR 이미지라면 true, visual 이미지라면 false입니다.

3. *reserved*
예약된 공간입니다.

BS2FaceEx

```
typedef struct {
    uint8_t       faceIndex;
    uint8_t       numOfTemplate;
    uint8_t       flag;
    uint8_t       reserved;

    uint32_t      imageLen;
    union {
        struct {
            uint16_t irImageLen;
            uint8_t  unused[6];            ///< 6 bytes (packing)
            uint8_t  imageData[BS2_MAX_WARPED_IMAGE_LENGTH];            ///<
40 * 1024 bytes
            uint8_t  irImageData[BS2_MAX_WARPED_IR_IMAGE_LENGTH];       ///<
30 * 1024 bytes
            BS2TemplateEx templateEx[BS2_MAX_TEMPLATES_PER_FACE_EX];    ///<
20 * 556 bytes
        };

        uint8_t   *rawImageData;
    };
} BS2FaceEx;
```

FaceStation F2

1. *faceIndex*
얼굴 인덱스입니다.

2. *numOfTemplate*
Visual, IR을 포함한 얼굴 템플릿 개수입니다.

3. *flag*
이미지가 WARP된 이미지인지 여부를 나타냅니다.
여기서 WARP란, 이미지에는 다양한 신체 부분이 담겨 있을 수 있는데, 이 중 얼굴 부분을 뽑아내는, 일종의 일반화라고 보시면 됩니다.
flag가 1로 설정되면, 장치는 union 내의 struct로 정의되어 있는 5개의 정보를 참조합니다.
flag가 0으로 설정되면, 장치는 union 내의 rawImageData를 참조합니다.
만약에 사용자가 WARP되지 않은 임의의 이미지를 가지고 얼굴을 등록하고자 하는 경우,
flag는 BS2_FACE_EX_FLAG_NONE(0)으로, rawImageData의 주소공간에 이미지 데이터로 설정하고, 이미지 데이터의 크기를 imageLen으로 지정해 주시면 됩니다.
이렇게 되면 장치는 rawImageData를 가지고 자동으로 WARP 과정을 거쳐서 struct 내의 정보들을 채워주게 될 것입니다.
rawImageData와 struct가 union으로 묶여있음을 주목해 주세요.

값	설명
BS2_FACE_EX_FLAG_NONE	0x00
BS2_FACE_EX_FLAG_WARPED	0x01
BS2_FACE_EX_FLAG_ALL	0xFF

4. *reserved*
예약된 공간입니다.

5. *imageLen*
얼굴 이미지 데이터 크기입니다.

6. *irImageLen*
IR 이미지 데이터 크기입니다.

7. *unused*
사용하지 않는 공간입니다. (packing용)

8. *imageData*
WARP 되어진 얼굴 이미지 데이터입니다. rawImageData로부터 자동으로 만들어집니다.

9. *irImageData*
IR 이미지 데이터입니다. WARP 단계에서는 IR 이미지는 존재하지 않을 수 있습니다.
인증을 시도하게되면 자동으로 생성됩니다.
10. *templateEx*
Visual 또는 IR 이미지에 대한 탬플릿 데이터입니다.

11. *rawImageData*
WARP되지 않은 이미지 정보입니다. (JPG 이미지만 지원, 최소크기 250 * 250)

From:
https://kb.supremainc.com/kbtest/ - **BioStar 2 Device SDK**

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=ko:face_api&rev=1636882473

Last update: **2021/11/14 18:34**