



LinkSquare API Reference for Stratio (iOS)

References and Examples (Stratio)

Version 1.0.0

Last updated: 2018-01-31

Contents

1	개요	3
2	LinkSquare API Reference	3
2.1	기본 동작	3
2.2	각 API 설명	3
2.2.1	Initialize.....	3
2.2.2	RegisterEventCallback.....	4
2.2.3	LSDeviceEventType	5
2.2.4	Connect.....	6
2.2.5	IsConnected	7
2.2.6	GetDeviceInfo	7
2.2.7	LSDeviceInfo	7
2.2.8	Scan.....	8
2.2.9	Close	8
2.2.10	GetLSError	9
2.2.11	LSFRAME.....	9
2.2.12	ChangePassword	10
2.2.13	FactoryReset.....	10
2.2.14	SetWlanInfo	11
2.2.15	SetAlias	11
2.3	에러 메시지 목록	12
3	LinkSquare Finder API Reference.....	13
3.1	기본 동작	13
3.2	API 설명	14
3.2.1	SmartConfig_Start.....	14
3.2.2	SmartConfig_Stop	14
3.2.3	LinkSquareFinder_RegisterCallback.....	14
3.2.4	LinkSquareDevice.....	15
3.2.5	LinkSquareFinder_Start	15
3.2.6	SmartConfig_Stop	16
4	기술 지원 안내.....	16

1 개요

이 문서는 스트라티오에서 사용되는 LinkSquare™ API for iOS 의 기능 및 사용 방법에 대해 설명합니다. LinkSquare API 는 LinkSquare™ 기기와 연결 및 데이터 수집 기능을 제공하는 소프트웨어 라이브러리입니다. LinkSquare API for iOS 는 iOS 앱 개발에 사용할 수 있는 LinkSquare API 입니다.

2 LinkSquare API Reference

2.1 개요

LinkSquareAPI for iOS 는 LinkSquare 기기에 접속하여 스캔하기 위한 기능을 제공합니다. 또한 LinkSquare 기기의 Alias 설정, Wi-Fi 정보설정 API, 기기 공장초기화 및 AP 모드의 Password 를 변경하기 위한 API 들을 제공합니다.

Xcode 로 개발 되는 iOS 프로젝트에서 참조 할 수 있도록 Framework Package 로 제공되며 C++를 사용하여 작성되었습니다. iOS 프로젝트에서는 확장자가 “.mm” 인 Objective-C++ 파일에서 사용할 수 있습니다. Objective-C 와 Swift 를 사용하는 프로젝트는 사용자가 Wrapper 작성이 필요 합니다.

2.2 기본 동작

LinkSquareAPI Framework 의 LinkSquareAPI/LSAPI_internal.h 헤더 파일에 정의되어 있는 Initialize 함수를 통해 LinkSquare 타입의 핸들을 생성한 다음 LSAPI 가 제공하는 함수를 통해 LinkSquare 기기와 연결하여 사용자가 필요한 작업을 진행 할 수 있습니다.

LinkSquare 기기와 Connect 함수를 통해 연결을 시도합니다.

연결이 성공하면 Scan 함수를 통해 LinkSquare 기기에서 스캔 데이터를 얻어 올 수 있습니다.

필요한 스캔 데이터 수집이 완료되면 Close 함수를 통해 LinkSquare 핸들을 정리하고 LinkSquare 기기와 연결을 종료 합니다.

2.3 각 API 설명

2.3.1 Initialize

Syntax

```
LinkSquare Initialize();
```

Return value

Type: LinkSquare

함수 호출에 성공하면 LinkSquare 핸들을 반환합니다. 실패 시 NULL 을 반환하며 GetLSLError 함수를 통해 자세한 오류 내용을 확인 할 수 있습니다.

Remark

Initialize 함수는 LSAPI 를 처음 사용하기 위한 초기화 작업을 진행합니다. LSAPI 사용시 제일 먼저 한 번 호출 되어야 합니다. 호출에 성공 시 반환되는 LinkSquare 핸들은 다음 LSAPI 호출을 위해 필요합니다.

2.3.2 RegisterEventCallback

Syntax

```
void RegisterEventCallback(LinkSquare h, DeviceEventCallbackFn func, void* userData);
```

Parameter

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

func:[in], Type: DeviceEventCallbackFn

LinkSquare 기기에서 버튼이 눌렸을 때 호출될 Callback 함수입니다.

userData:[in], Type: (Void*)

Callback 함수 호출 시 같이 전달될 사용자 정의 데이터 입니다.

Remakr

사용자가 LinkSquare 기기의 스캔 버튼을 눌렀을 때 등록된 Callback 함수를 LinkSquare API 에서 호출합니다. Callback 함수가 등록되지 않다면 스캔 버튼 이벤트는 무시 됩니다.

Callback 함수가 호출되면 Scan 함수를 호출 하여 스캔을 진행 하여야 합니다

DeviceEventCallbackFn 타입은 아래와 같습니다

```
typedef void(LSAPI_CALLBACK *DeviceEventCallbackFn)(void* userData, LSDeviceEventType type, int value);
```

userData 는 Callback 함수등록 시 사용자가 입력한 userData 로 Callback 함수가 호출 될 때 파라미터로 사용됩니다. 사용자가 필요로 하는 정보를 (Void*)타입으로 전달한 뒤 Callback 함수에서 원래의 타입으로 변환한 뒤 사용 가능합니다. LSDeviceEventType enum 을 Parameter 로 받습니다. LSDeviceEventType 에 대한 자세한 내용은 항목을 참고하세요.

C++ 로 되어 있는 LinkSquare API 를 통해 Objective-C 의 객체를 정확히 참조 하기 위해서는 캐스팅 때 `__bridge` 키워드를 사용하여야 합니다. 자세한 사용 예제는 LinkSquare API Walkthrough (iOS) 문서를 참고하세요.

LSDeviceEventType 에 대한 자세한 내용은 항목을 참고하세요.

예시) Callback 함수 정의

```
// LinkSquare 기기 버튼 눌렀을 때 Callback
void LSAPI_CALLBACK DeviceCallback(void* userData, LSDeviceEventType type, int value)
{
    if (type == LSDeviceEventType::ButtonEvent)
    {
        printf("Scan Button Pressed.\n");
        // 스캔
        // LED 광원으로 스캔할 프레임 수
        int numLEDFrames = 3;

        // Blub 광원으로 스캔할 프레임 수
        int numBulbFrames = 3;

        // 스캔 결과를 받아올 vector
        vector<LSFRAME*> frames;
        int ret = LSAPI::Scan(g_hLS, numLEDFrames, numBulbFrames, frames);
        if (ret != 1) {
            cout << LSAPI::GetLSError() << endl;
            LSAPI::Close(g_hLS);
            return;
        }

        // 스캔 받은 데이터를 출력
        for (size_t i = 0; i < frames.size(); i++) {
            printf("Frame %d: width = %d, height = %d, light_source = %hu\n",
                (int)i, frames[i]->width, frames[i]->height, frames[i]->light_source);
        }
    }
    else if (type == LSDeviceEventType::TimeoutEvent)
    {
        // LinkSquare기기와의 네트워크 연결 제한 시간이 초과하였습니다.
        printf("Device Network Timeout.\n");

        // LinkSquare Handle의 적절한 해제를 위해 Close를 호출 하세요.
        LSAPI::Close(g_hLS);
    }
    else if (type == LSDeviceEventType::NetworkCloseEvent)
    {
        // LinkSquare기기와의 네트워크 연결이 끊겼습니다.
        printf("Device Network Closed.\n");

        // LinkSquare Handle의 적절한 해제를 위해 Close를 호출 하세요.
        LSAPI::Close(g_hLS);
    }
}
}
```

2.3.3 LSDeviceEventType

```
typedef enum _eventType{
    ButtonEvent = 0,
    TimeoutEvent = 1,
    NetworkCloseEvent = 2,
} LSDeviceEventType;
```

Remark

RegisterEventCallback 으로 등록된 DeviceEventCallbackFn 함수의 인자로 사용됩니다.

값	의미
ButtonEvent	사용자가 LinkSquare 기기의 스캔 버튼을 눌렀을 때 Callback 함수를 호출시에 사용됩니다. Callback 함수가 호출되면 Scan 함수를 통해 Scan 을 진행 하여야 합니다.
TimeoutEvent	일정 시간 이상 LinkSquare 기기와 네트워크 전송이 없을 때 Callback 함수를 호출시에 사용됩니다. Callback 함수가 호출되면 사용자에게 적절한 오류 메시지를 표시하고 Close 함수를 호출 하여 LinkSquare 기기와의 연결을 해지하고 LinkSquare 핸들을 정리 해야 합니다.
NetworkCloseEvent	LinkSquare 와의 네트워크 연결이 끊어졌을 때 Callback 함수를 호출하며 사용됩니다. Callback 함수가 호출되면 사용자에게 적절한 오류 메시지를 표시하고 Close 함수를 호출 하여 LinkSquare 기기와의 연결을 해지하고 LinkSquare 핸들을 정리 해야 합니다.

2.3.4 Connect

Syntax

```
int Connect(LinkSquare h, const char* IPAddr, int port);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

IPAddr:[in], Type: const char*

접속할 LinkSquare 기기의 IP 주소의 문자열입니다.

Port:[in], Type: int

접속할 LinkSquare 기기의 포트 번호 입니다.

Return value

연결에 성공하면 1 을 반환합니다. 함수 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLError 함수를 통해 확인 할 수 있습니다.

Remark

전달되는 LinkSquare 핸들 값은 반드시 Initialize 함수의 반환 값을 사용하여야 합니다. Initialize 실패 시 Connect 함수 호출도 실패 합니다. IPAddr 은 \0 으로 끝나는 문자열이어야 합니다. IPAddr 과 port 파라미터는 유효한 값이어야 합니다.

Connect 함수 반환까지 호출은 Block 됩니다.

2.3.5 IsConnected

Syntax

```
bool IsConnected(LinkSquare h);
```

Parameter

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

Return value

LinkSquare 기기와 연결이 되어 있으면 true 를 반환합니다. 연결이 되어 있지 않다면 false 를 반환합니다.

2.3.6 GetDeviceInfo

Syntax

```
int GetDeviceInfo(LinkSquare h, LSDeviceInfo * deviceInfo);
```

Parameter

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

deviceInfo:[in], Type: (LSDeviceInfo*)

현재 연결된 LinkSquare 기기정보를 담을 구조체입니다. LSDeviceInfo 에 대한 자세한 내용은 항목을 참고하세요.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLSError 함수를 통해 확인 할 수 있습니다.

Remark

현재 연결되어 있는 LinkSquare 기기에 대한 자세한 정보를 얻기 위한 함수 입니다. 호출에 성공하면 LSDeviceInfo 구조체에서 정보를 확인 할 수 있습니다.

2.3.7 LSDeviceInfo

Syntax

```
typedef struct {  
    const char* SWVersion;  
    const char* HWVersion;  
    const char* DeviceID;  
    const char* Alias;  
    const char* OPMODE;  
} LSDeviceInfo;
```

Members

SWVersion: 현재 접속 중인 LinkSquare 기기의 Software 버전 정보 문자열 입니다.

HWVersion: 현재 접속 중인 LinkSquare 기기의 Hardware 버전 정보 문자열입니다.

DeviceID: 현재 접속 중인 LinkSquare 기기의 Unique Device ID(UDID) 정보 문자열입니다.

Alias: 현재 접속 중인 LinkSquare 기기의 이름입니다.

OPMode: 현재 접속 중인 LinkSquare 기기의 작동 모드 정보입니다. AP 모드일 경우 “AP”, IoT 모드일 경우 “STATION” 값을 가집니다.

2.3.8 Scan

Syntax

```
int Scan(LinkSquare h, int numLEDFrames, int numBulbFrames, std::vector<LSFRAME*>& frames);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

numLEDFrames[in], Type: int

LED 조명으로 스캔 될 프레임 수 입니다. 최소 1 에서 최대 8 사이의 정수 값입니다.

numBulbFrames[in], Type: int

Bulb 조명으로 스캔 될 프레임 수 입니다. 최소 1 에서 최대 8 사이의 정수 값입니다.

frames:[in], Type: std::vector<LSFRAME*>

LinkSquare 기기에서 스캔 된 데이터를 반환 받을 LSFRAME 구조체의 포인터의 std::vector 컨테이너입니다.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLSError 함수를 통해 확인 할 수 있습니다.

Remark

Scan 함수는 LinkSquare 기기에 스캔 명령을 전달하고, 실제 스캔 데이터를 전송 받습니다. 스캔 함수 호출은 모든 데이터를 전송 받을 때까지 Block 됩니다.

frames 를 통해 반환되는 스캔 데이터들은 다음 번 Scan 함수 호출 혹은 Close 함수 호출 전까지 유효합니다. 필요시 LSFRAME 구조체의 데이터들을 복사하여 사용하여야 합니다.

2.3.9 Close

Syntax

```
void Close(LinkSquare& h);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

Remark

LSAPI 사용을 중단하기 위해 LinkSquare 와 연결을 해지하고 내부에서 사용하는 리소스들을 정리하기 위한 함수 입니다.

2.3.10 GetLSError

Syntax

```
const char * GetLSError();
```

Return value

가장 최근 발행한 오류에 대한 상세한 메시지를 반환합니다.

Remark

LSAPI 내부에서 발생한 가장 최근 오류에 대한 상세한 메시지를 반환합니다.

디버깅 뿐만 아니라 사용자에게 오류를 표시하기 위해 사용할 수 있습니다.

메시지 종류와 의미는 에러 메시지 목록 항목을 참고하세요.

2.3.11 LSFRAME

Syntax

```
typedef struct {  
    int length;  
    unsigned char light_source;  
    int frame_no;  
    unsigned short *raw_data;  
    float *data;  
} LSFRAME;
```

Members

length: LinkSquare 기기에서 전달 받은 스캔 데이터의 길이 입니다. 기본 값은 600 입니다.

light_source: 현재 프레임의 광원 정보입니다. LED 는 0, Bulb 는 1 입니다.

frame_no: 각 스캔 당 고유 프레임 번호입니다. 첫 프레임은 1 이고, 프레임 마다 1 씩 증가합니다.

raw_data: 스캔 데이터의 원본 데이터 입니다. Unsigned short 형 1 차원 배열의 포인터 입니다. 배열의 길이는 length 입니다.

data: 원본 데이터를 프로세싱 한 데이터 입니다. float 형 배열의 포인터 입니다. 배열의 길이는 length 입니다.

Remark

LinkSquare 기기에서 전달 받은 스캔데이터는 LSFRAME 구조체에 저장되어 사용됩니다. 스캔데이터는 기본 600 개의 값을 갖는 1 차원 데이터입니다. LSFRAME 에는 원본 데이터를 저장한 raw_data 와 이 원본 데이터를 프로세싱한 data 두 멤버를 통해 값을 얻을 수 있습니다. 스캔데이터는 배열 인덱스에 따라 (600)을 따라서 빛의 파장이 달라집니다. 데이터를 참조하는 방법은 아래의 예제코드를 참고하세요.

예제) 모든 raw_data 접근 코드 조각

```
LSFRAME *frame = frames[i];  
for (int index = 0; index < frame->length; index++) // 길이 0...599  
{  
    auto value = frame->raw_data[index]; // 모든 원본 데이터 확인.  
}
```

2.3.12 ChangePassword

Syntax

```
int ChangePassword(LinkSquare h, string oldpw, string newpw);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

oldpw[in], Type: string

현재 사용 중인 암호입니다.

newpw[in], Type: string

새로 바뀔 암호입니다.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLSError 함수를 통해 확인 할 수 있습니다.

Remark

LinkSquare 기기의 AP 접속을 위한 암호를 변경하는 함수 입니다. AP 암호의 초기 값은 "00000000" 입니다. ChangePassword 함수는 Connect 함수를 먼저 호출 하여 LinkSquare 기기에 연결 된 다음 호출 하여야 합니다.

2.3.13 FactoryReset

Syntax

```
int FactoryReset (LinkSquare h);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLSError 함수를 통해 확인 할 수 있습니다.

Remark

LinkSquare 기기에게 공장 초기화 과정을 진행 하도록 합니다. 기기의 펌웨어가 공장 출하시 버전으로 초기화 됩니다.

2.3.14 SetWlanInfo

Syntax

```
int SetWlanInfo(LinkSquare h, string ssid, string password, uint8_t securityOption);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

ssid:[in], Type: std::string

LinkSquare 기기가 IoT 모드에서 접속할 AP 의 SSID 입니다. 최대 31 자 입니다.

password:[in], Type:std::string

LinkSquare 기기가 IoT 모드에서 접속할 AP 의 비밀번호 입니다. 최대 62 자 입니다.

securityOption:[in], Type:uint8_t (unsigned char)

LinkSquare 기기가 IoT 모드에서 접속할 AP 의 보안 정보입니다. 0: Open, 1: WEP, 2:WPA/WPA2 입니다.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLSError 함수를 통해 확인 할 수 있습니다.

Remark

LinkSquare 기기가 IoT 모드로 동작할 때 접속할 AP 정보를 설정하는 함수입니다.

2.3.15 SetAlias

Syntax

```
int SetAlias(LinkSquare h, string alias);
```

Parameters

h:[in], Type: LinkSquare

Initialize 함수에서 반환된 LinkSquare 핸들 값입니다.

alias:[in], Type: std::string

LinkSquare 기기에 설정할 이름입니다. 최소 4 자 이상 최대 16 자 입니다.

Return value

호출 성공 시 1 을 반환합니다. 호출에 실패 시 1 이외의 값을 반환합니다. 자세한 오류 내용은 GetLError 함수를 통해 확인 할 수 있습니다.

Remark

LinkSquare 기기의 이름을 설정하는 함수입니다. 이름은 영문자와 숫자 특수문자의 조합으로 길이는 최소 4~ 최대 16 자입니다. 설정된 이름은 AP 모드시 "LS1-" 뒤에 붙여 SSID 로 사용됩니다.

IoT 모드에서는 Find Device 시 표시되는 이름으로 사용됩니다.

2.4 에러 메시지 목록

LinkSquare API 호출 시 오류가 발생 했을 때 GetLError 함수에서 반환되는 메시지의 목록입니다.

에러 메시지	발생 이유
Failed to allocate memory.	메모리 부족으로 내부에서 사용되는 메모리 할당에 실패 하였을 때 출력합니다.
Failed to encrypt data.	데이터의 암호화 실패 시 출력 됩니다.
Internal error occurred.	API 내부 로직 문제 혹은 알 수 없는 이유로 오류가 발생 하였을 때 출력 됩니다.
Invalid data received.	LinkSquare 에서 네트워크로 전달 받은 데이터에 문제가 있을 때 출력 됩니다.
Invalid LinkSquare handle.	API 호출 시 잘못된 LinkSquare Handle 을 전달 시 출력됩니다.
Invalid Port value.	API 호출 시 잘못된 Port 값을 전달 했을 때 출력됩니다.
LinkSquare returns error, %s	LinkSquare 에서 오류 응답을 받았을 때 출력됩니다.
Network Connection Failed.	LinkSquare 와 네트워크 연결, 데이터 전송 및 수신에 실패 시 출력됩니다.
Not connected LinkSquare.	API 호출 시 Connect 안된 상태의 LinkSquare Handle 을 사용 시 출력됩니다.
Only LinkSquare devkit user can access to this device.	개발자킷용 LinkSquare 가 아닌 장비를 사용시 출력됩니다.
Request timed out. Unable to receive a response.	네트워크 혹은 다른 문제로 제한 시간안에 응답을 못 받았을 때 출력됩니다.
Too many number of Bulb frames value.	API 호출 시 유효하지 않은 값을 전달 했을 때 출력됩니다.
Too many number of LED frames value.	API 호출 시 유효하지 않은 값을 전달 했을 때 출력됩니다.
Too small number of Bulb frames value.	API 호출 시 유효하지 않은 값을 전달 했을 때 출력됩니다.
Too small number of LED frames value.	API 호출 시 유효하지 않은 값을 전달 했을 때 출력됩니다.

Unable to connect to LinkSquare.

LinkSquare 장비에 연결 할 수 없는 경우 출력됩니다.

3 LinkSquare Finder API Reference

3.1 기본 동작

LinkSquare Finder API 는 IoT 모드의 LinkSquare 기기에게 네트워크 정보를 전달하는 SmartConfig 기능과 네트워크에 접속한 LinkSquare 기기를 검색하는 Finder 기능을 제공합니다.

SmartConfig 서비스는 LinkSquare 기기가 Wi-Fi 접속에 필요한 SSID 와 Password 정보를 전달하기 위한 서비스로 사용자에게 입력 받은 SSID 와 Password 그리고 현재 네트워크의 Gateway IP 를 입력 받습니다.

SSID 와 Password 를 정해진 방법으로 인코딩하여 패킷을 생성하고 패킷을 Wi-Fi 네트워크에 특정기기(Gateway)에 전달하게 되면 LinkSquare 에서 해당 패킷의 정보를 수신하고 SSID 와 Password 정보를 디코딩하여 AP 에 접속을 합니다.

Finder 서비스는 네트워크에 접속한 LinkSquare 기기를 찾기 위한 서비스 입니다. IoT 모드로 AP 에 접속한 LinkSquare 기기는 mDNS 를 이용한 DNS-SD(Service Discovery)를 사용하여 자신의 IP, Port 와 별칭(Alias)를 네트워크에 전송합니다. Finder 서비스는 이 정보를 받아 사용자에게 네트워크에서 새로 발견된 기기 정보를 전달합니다.

한 네트워크에 여러 대의 LinkSquare 기기가 있을 경우 발견되는 순서에 따라

LinkSquareFinder_RegisterCallback 함수로 등록된 FoundNewDeviceFunc 함수로 Callback 이 호출됩니다.

이때 인자로 LinkSquareDevice 객체가 전달되며 이를 통해 IP, Port, Alias 정보를 얻을 수 있습니다.

IP 와 Port 정보를 사용하여 해당 LinkSquare 기기에 접속할 수 있습니다. 사용자에게는 Alias 이름을 표시하여 사용자가 어떤 LinkSquare 기기에 접속할지 알 수 있도록 해야 합니다.

LinkSquareFinder 서비스를 시작할 때는 일반 사용자의 편의를 위해 SmartConfig, Finder 두 서비스를 동시에 시작합니다.

서비스가 시작한 다음 일정시간 (1 분정도)이 지나면 서비스를 중지하고 만약 시간동안 새로운 기기를 찾지 못하였다면 사용자에게 안내를 해야 합니다.

3.2 API 설명

3.2.1 SmartConfig_Start

Syntax

```
int SmartConfig_Start(std::string ssid, std::string key, std::string gatewayIp);
```

Parameter

ssid:[in], Type: std::string

LinkSquare 기기에 전달할 네트워크의 SSID 입니다. 최대 31 자 입니다.

key:[in], Type: std::string

LinkSquare 기기에 전달할 네트워크의 Password 입니다. 최대 31 자 입니다.

gatewayIp:[in], Type: std::string

현재 접속 중인 Wi-Fi 네트워크의 Gateway IP 입니다.

Return value

연결에 성공하면 1 을 반환합니다. 함수 호출에 실패 시 1 이외의 값을 반환합니다

Remakr

SmartConfig 는 IoT 모드의 LinkSquare 기기가 네트워크에 접속하기 위해 필요한 네트워크의 SSID 와 Password 를 전달하기 위한 기능입니다. Gateway IP 로 인코딩 된 패킷을 UDP 로 전송하고 LinkSquare 기기는 Wi-Fi 전파를 수신하여 정보를 얻습니다. SmartConfig 는 2.4Ghz 의 Wi-Fi 네트워크에서만 동작하고 802.11 g/b/n 등을 지원합니다. 이중 802.11n 일 경우 제약이 있습니다.

3.2.2 SmartConfig_Stop

Syntax

```
int SmartConfig_Stop();
```

Return value

연결에 성공하면 1 을 반환합니다. 함수 호출에 실패 시 1 이외의 값을 반환합니다

Remakr

SmartConfig 서비스를 중지 합니다.

3.2.3 LinkSquareFinder_RegisterCallback

Syntax

```
void LinkSquareFinder_RegisterCallback(FoundNewDeviceFunc func);
```

Parameter

func:[in], Type: FoundNewDeviceFunc

네트워크에서 새로운 LinkSquare 기기를 발견 시 Callback 될 함수를 등록합니다.

Remakr

LinkSquare Finder 서비스에서 새로운 LinkSquare 기기를 발견 시 등록된 FoundNewDeviceFunc 함수 타입을 호출합니다. FoundNewDeviceFunc 은 LinkSquareDevice 객체를 인자로 전달 받습니다.

```
typedef void(_CALLBACK *FoundNewDeviceFunc)(std::shared_ptr<LinkSquareDevice>
newDevice);
```

3.2.4 LinkSquareDevice

Syntax

```
class LinkSquareDevice
{
public:
    std::string FullName = "";
    std::string IP = "";
    uint16_t Port = 0;
    std::string Alias = "";
};
```

Members

FullName: LinkSquare 기기의 네트워크 상의 DNS 이름입니다.

IP: LinkSquare 기기의 IP 주소 입니다.

Port: LinkSquare 기기의 Port 번호 입니다.

Alias: LinkSquare 기기의 이름입니다.

Remakr

사용자에게는 Alias 이름을 표시해주어야 합니다. LinkSquare 기기와 접속은 IP 와 Port 정보를 사용합니다.

3.2.5 LinkSquareFinder_Start

Syntax

```
int LinkSquareFinder_Start();
```

Return value

연결에 성공하면 1 을 반환합니다. 함수 호출에 실패 시 1 이외의 값을 반환합니다.

Remakr

Finder 서비스를 시작합니다. Finder 서비스는 네트워크에 접속한 LinkSquare 기기가 전송하는 mDNS 패킷을 수신하여 LinkSquare 기기를 검색합니다. IoT 모드의 LinkSquare 는 네트워크에 성공적으로 접속 했을 때 자신의 이름과 IP 정보 등을 mDNS 형식으로 네트워크상에 Multicast 합니다. Finder 서비스는 이를 수신하여 새로운 LinkSquare 기기가 발견되면 등록된 Callback 함수를 호출 합니다. 이를 위해 UDP 5353 포트를 사용합니다.

3.2.6 SmartConfig_Stop

Syntax

```
int LinkSquareFinder_Start();
```

Return value

연결에 성공하면 1 을 반환합니다. 함수 호출에 실패 시 1 이외의 값을 반환합니다.

Remakr

LinkSquare Finder 서비스를 중지 합니다.

4 기술 지원 안내

- LinkSquare Professional 기술 지원

LinkSquare Professional 제품에 관련 문의 사항은 아래 연락처로 문의 하십시오.

전화: 02 - 6205 - 7456

전자메일: contact@stratiotechnology.com

웹 사이트: <http://linksquare.io/contact.html>