



AULORA API

오로라 LoRa API



Version 1.2.1

2018 DECEMBER 11

세타랩 주식회사
help@setalab.com

AuLoRa API

목차

1. 디바이스 API	2
2. LORA 서버 API	5
3. 기타 API	14

디바이스 API

1. 디바이스 API : 4 개

DevOpen : 디바이스의 시리얼 Comport 를 설정하고, 연결을 한다

DevClose : 연결된 디바이스를 해제한다

DevSetATCmd : AT 커맨드 명령어를 보낸다

DevGetATCmd : 디바이스로부터 AT 커맨드 값을 가져온다

2. DevOpen

디바이스의 시리얼 Comport 를 설정하고, 연결을 한다

- BOOL DevOpen(int portnumber, int bitrate)

Input parameter

Int portnumber : 디바이스를 PC 에 연결했을 때 생성되는 comport 번호를 입력한다

Int bitrate : 디바이스에 맞는 data bitrate 을 입력한다. (9600/38400/115200)

Result

True : 포트 연결 성공

False : 포트 연결 실패

사용예제

```
Var retValue = DevOpen(6, 38400);
```

3. DevClose

연결된 디바이스의 시리얼 Comport 를 해제한다

- BOOL DevClose()

Input parameter

없음

Result

True : 포트 해제 성공
 False : 포트 해제 실패

사용예제

DevClose();

4. DevSetATCmd

AT 커맨드 명령어를 보낸다

- CString DevSetATCmd(CString strcmd)

Input parameter

CString strcmd : 디바이스에 보낼 명령어를 셋팅한다. "AT+SEND 021234"

Result

error : 커맨드가 정상적으로 실행되지 않은 경우

사용예제

DevSetATCmd ("AT+SEND 1234");

5. DevGetATCmd

디바이스로부터 AT 커맨드를 이용하여 값을 읽어 들인다.

- CString DevGetATCmd(CString strcmd)

Input parameter

CString strcmd : 디바이스에 보낼 명령어를 셋팅한다. "AT+AKEY=?"

Result

error : 커맨드가 정상적으로 실행되지 않은 경우

string : 커맨드가 정상 실행되어 요청한 값이 들어온다.

사용예제

Var retStr = DevGetATCmd ("AT+AKEY=?"); //retStr = 0000000000000000

6. 사용 예제

```
function main()
{
    SendToLog("Comport 6 번, baudrate 38400");
    DevOpen(6, 38400);
    SetWait(1000);

    SendToLog("기기의 Akey 값을 설정합니다. ")
    DevSetATCmd("AT+AK 00000000000000000000000000000000");
    SetWait(100);

    SendToLog("기기의 Akey 값을 읽습니다")
    retval = DevGetATCmd("AT+AK=?");
    SendToLog("%s", retval);

    DevClose();
}
```

LORA API

1. LoRa API : 12 개

SrvOn
SrvOff
SrvTxOn
SrvTxOff
SrvSetConfirmed
SrvSetMacData
SrvSetAppData
SrvSetParam
SrvGetParam
SrvGetState
SrvRx2Transmit //새로 추가됨
SrvRx2DownNow //새로 추가됨

2. SrvOn

로라 서버를 실행하여 Gateway 와 데이터를 주고 받는 상태가 된다.

- BOOL SrvOn()

Input parameter

없음

Result

True : 서버 프로그램 실행 성공

False : 서버 프로그램 실행 실패

사용예제

```
var ret= SrvOn();
```

3. SrvOff

로라 서버를 닫는다.

- BOOL SrvOff()

Input parameter

없음

Result

True : 서버 프로그램 종료 성공

False : 서버 프로그램 종료 실패

사용예제

```
Var ret= SrvOff();
```

4. SrvTxOn

서버에서 게이트웨이로 TX 커맨드를 송출할 수 있게 한다.

- BOOL SrvTxOn()

Input parameter

없음

Result

True : 서버 TX 커맨드 송출 설정 성공

False : 서버 TX 커맨드 송출 설정 실패

사용예제

```
Var ret= SrvTxOn();
```

5. SrvTxOff

서버에서 게이트웨이로 TX 커맨드를 송출할 수 없게 한다. 주로 retry 테스트 시 사용한다.

- BOOL SrvTxOff()

Input parameter

없음

Result

True : 서버 TX 커맨드 Block 설정 성공

False : 서버 TX 커맨드 Block 설정 실패

사용예제

```
Var ret= SrvTxOff();
```

6. SrvSetConfirmed

서버에서 기기로 데이터를 보낼 때 confirmed 로 보낼지 unconfirmed 로 보낼 지 결정한다.

- BOOL SrvSetConfirmed(BOOL bConfirmed)

Input parameter

BOOL bConfirmed : TX 에서 보낼 데이터를 confirmed 로 보낼지 unconfirmed 로 보낼지 설정한다

true (1) : confirmed

false (0): unconfirmed

Result

True : TX message confirmed 설정 성공

False : TX message confirmed 설정 실패

사용예제

```
Var ret= SrvSetConfirmed(true); //confirmed 설정
```

7. SrvSetMacData

서버에서 디바이스로 보낼 TX 데이터 중 LoRa Mac 커맨드를 설정한다. 로라 ClassA 의 특성상, 해당 API 를 사전에 예약한 후 디바이스에서 데이터를 보낼 때까지 대기하고 있다가 디바이스로부터 데이터를 받으면 그 이후 실행된다.

- BOOL srvSetMacData(CString strType, CString strData)

Input parameter

CString strType : LoRa 스펙 문서에 정의된 Mac Data Type 과 일치한다.

CID	Command	Transmitted by		Short Description
		End-device	Gateway	
0x02	LinkCheckReq	x		Used by an end-device to validate its connectivity to a network.
0x02	LinkCheckAns		x	Answer to LinkCheckReq command. Contains the received signal power estimation indicating to the end-device the quality of reception (link margin).
0x03	LinkADRReq		x	Requests the end-device to change data rate, transmit power, repetition rate or channel.
0x03	LinkADRAns	x		Acknowledges the LinkRateReq.
0x04	DutyCycleReq		x	Sets the maximum aggregated transmit duty-cycle of a device
0x04	DutyCycleAns	x		Acknowledges a DutyCycleReq command
0x05	RXParamSetupReq		x	Sets the reception slots parameters
0x05	RXParamSetupAns	x		Acknowledges a RXSetupReq command
0x06	DevStatusReq		x	Requests the status of the end-device
0x06	DevStatusAns	x		Returns the status of the end-device, namely its battery level and its demodulation margin
0x07	NewChannelReq		x	Creates or modifies the definition of a radio channel
0x07	NewChannelAns	x		Acknowledges a NewChannelReq command
0x08	RXTimingSetupReq		x	Sets the timing of the of the reception slots
0x08	RXTimingSetupAns	x		Acknowledges RXTimingSetupReq command
0x09	TxParamSetupReq		x	Used by the network server to set the maximum allowed dwell time and Max EIRP of end-device, based on local regulations
0x09	TxParamSetupAns	x		Acknowledges TxParamSetupReq command
0x0A	DIChannelReq		x	Modifies the definition of a downlink RX1 radio channel by shifting the downlink frequency from the uplink frequencies (i.e. creating an asymmetric channel)
0x0A	DIChannelAns	x		Acknowledges DIChannelReq command

CString strData : 각 Mac Type 에 맞는 Data 값을 입력한다. 자세한 Data 값은 로라 스펙 문서를 참조할 것

Result

True : Mac Data 설정 성공

False : Mac Data 설정 실패

사용예제

```
Var ret= SrvSetMacData(0x03, "41 07 00 61"); //Link ADR Req
```

```
Ret = SrvSetMacData(0x06, ""); // Device Status Req 파라미터가 없음
```

8. SrvSetAppData

서버에서 디바이스로 보낼 TX 데이터 중 실제 Data 부분이다.

- BOOL SrvSetAppData(int port, CString strData)

Input parameter

Int port : Application Data port 로 0x1 ~ 0xDD 까지 사용 가능하다.

CString strData : 디바이스에게 보낼 실제 데이터를 셋팅한다.

Result

True : App Data 설정 성공

False : App Data 설정 실패

사용예제

```
SrvSetAppData(0x02, "11 22 33 44 55"); //2 번 포트로 1122334455 를 보낸다
```

9. SrvSetParam

서버에 디바이스와 관련된 값을 설정한다. 처음 서버를 실행한 후에 사용된다.

- BOOL SrvSetParam(CString strKey, CString strValue)

Input parameter

CString strKey: 서버에 설정되어야 할 주요 키 값을 넣는다.

```
#define KEY_DEVADDR_ID           "DEVADDR"  
#define KEY_NETWORK_ID         "NETWORKID"  
#define KEY_APPNONCE_ID        "APPNONCE"  
#define KEY_APPKEY_ID          "APPKEY"  
#define KEY_APPREALKEY_ID      "APPREALKEY"  
#define KEY_NETSKEY_ID         "NETSKEY"  
#define KEY_APPSKEY_ID         "APPSKEY"  
#define KEY_CFLIST_ID          "CFLIST"  
#define KEY_TXDELAY_ID         "TXDELAY"  
#define KEY_RX2CHAN_ID         "RX2CHAN"  
#define KEY_SERVER_STATE_ID    "SERVERSTATE"  
#define KEY_DOWN_FCNT          "DOWNFCOUNT" //새로 추가됨  
#define KEY_JOINDL_ID          "JOINDL" // 새로 추가됨  
#define KEY_JOINRXDELAY_ID     "JOINRXDELAY" //새로 추가됨  
#define KEY_DATARATE_ID        "DATARATE" // 새로 추가됨  
#define KEY_RXDATARATE_ID      "RXDATARATE" //새로 추가됨
```

```

#define KEY_TXPOWER_ID          "TXPOWER"    // 새로 추가됨
#define KEY_FREQ_ID            "FREQ"        // 새로 추가됨

```

CString strValue : 각 키 값에 넣을 실제 값을 입력한다.

Result

True : 설정 성공

False : 설정 실패

사용예제

```

function main()
{
    var bret = SrvOn();
    SetWait(5000);
    SrvSetParam("DEVADDR",          "48000000");
    SrvSetParam("NETWORKID",       "240000");
    SrvSetParam("APPNONCE",        "010203");
    SrvSetParam("APPKEY",          "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00");
    SrvSetParam("APPREALKEY",      "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00");
    SrvSetParam("NETSKEY",         "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00");
    SrvSetParam("APPSKEY",         "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00");
    SrvSetParam("CFLIST",          "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00");
    SrvSetParam("DOWNFCOUNT",      "7F52"); // 0x7F52
    SrvSetParam("JOINDL",          "23");      // RX1 DR Offset=2, RX2 DR=3
    SrvSetParam("JOINRXDELAY",     "09"); // Delay 설정 0 ~ 0F
    SrvSetParam("DATARATE",        "DR0"); // DR0 ~ DR5
    SrvSetParam("DATARATE",        "SF12"); // SF12, SF11 ~ SF7
    SrvSetParam("DATARATE",        "0"); // RX datarate = TX datarate 동일하게 하기
    SrvTxOn();
    SetWait(2000);
    LoraMaxQueueCount(5);
    SetWait(1000);
    SrvSetParam("TXPOWER", "10"); // TxPower 를 10dBm 로 설정
    SrvSetParam("FREQ", "922.70"); // Frequency 를 922.70 으로 설정- 소수점 둘째자리
}

```

10. SrvGetParam

현재 서버에 저장된 디바이스와 관련된 값을 읽는다.

- CString SrvGetParam(CString strKey)

Input parameter

CString strKey: 서버에서 읽으려고 하는 키 값을 넣는다.

```
#define KEY_DEVADDR_ID           "DEVADDR"
#define KEY_NETWORK_ID          "NETWORKID"
#define KEY_APPNONCE_ID         "APPNONCE"
#define KEY_APPKEY_ID           "APPKEY"
#define KEY_APPREALKEY_ID       "APPREALKEY"
#define KEY_NETSKEY_ID          "NETSKEY"
#define KEY_APPSKEY_ID          "APPSKEY"
#define KEY_CFLIST_ID           "CFLIST"
#define KEY_TXDELAY_ID          "TXDELAY"
#define KEY_RX2CHAN_ID          "RX2CHAN"
#define KEY_SERVER_STATE_ID     "SERVERSTATE"
#define KEY_DOWN_FCNT           "DOWNFCOUNT" //추가되었지만 동작안함
#define KEY_JOINDL_ID           "JOINDL" //새로 추가됨
#define KEY_JOINRXDELAY_ID      "JOINRXDELAY" //새로 추가됨
#define KEY_DATARATE_ID         "DATARATE" //새로 추가됨
#define KEY_RXDATARATE_ID       "RXDATARATE" //새로 추가됨
#define KEY_RSSI_ID             "RSSI" //새로 추가됨
#define KEY_SNR_ID              "SNR" //새로 추가됨
#define KEY_FREQ_ID             "FREQ" //새로 추가됨
```

Result

error: 읽기 실패

Strval : 키에 저장된 값

사용예제

```
function main()
{
    var strVal;
    strVal = SrvGetParam("APPKEY");
    SendToLog("Application Key =" + strVal );

    strVal = SrvGetParam("NETSKEY");
    SendToLog("Network Session Key =" + strVal );
```

```

strVal = SrvGetParam("APPSKEY");
SendToLog("Application Session Key =" + strVal );

strVal = SrvGetParam("RXDATARATE"); // RX datarate 가져오기
SendToLog("Rx Datarate =" + strVal);

var rssi = SrvGetParam("RSSI");
SendToLog("RSSI =" +rssi);

var freq = SrvGetParam("FREQ");
if(freq == "922.90"){ //소수점 둘째자리까지 체크
    SendToLog("Current Channel = " +freq);
}
}

```

11. SrvGetState

현재 디바이스와 서버의 연결 상태를 리턴한다.

- CString SrvGetState ()

Input parameter

없음

Result

error: 읽기 실패

Strval : 서버와 단말기의 연결 상태를 리턴한다.

None : 연결되지 않음

Pseudo Joined

RealKey0

RealKey2

Real Joined

사용예제

```

function main()
{

```

```

var strserverstate = SrvGetState();
SendToLog("State =" + strserverstate);
if( strserverstate == "RealJoined"){
    SendToLog("JOIN SUCCESS");
}
}

```

12. SrvRx2Transmit (*추가됨)

서버에서 Rx2 전송을 하겠다고 설정해놓는 함수

- BOOL SrvRx2Transmit(BOOL rx2used)

Input parameter

BOOL rx2used (0) : 사용 안 함

BOOL rx2used (1) : 사용함

Result

True : 설정 성공

False : 설정 실패

사용예제

SrvRx2Transmit(1); //0 : 사용 안 함, 1: 사용함

13. SrvRx2DownNow (*추가됨)

Class C 인 경우 서버에서 Rx2 를 바로 보내는 함수

- BOOL SrvRx2DownNow()

Input parameter

없음

Result

True : 보내기 성공

False : 보내기 실패

사용예제

```
DevSetATCmd("AT+CLS C"); //디바이스 클래스를 C 로 세팅  
DevSetATCmd("AT+SCFG"); // configuration 을 저장  
SrvRx2DownNow() //RX2 보내기
```

기타 API

1. 기타 API : 5 개

SetWait
SendToLog
ToolQuit
SrvSetLogMode //새로 추가됨
SrvSetRFLogMode //새로 추가됨

2. SetWait

스크립트 내에서 명령어와 명령어 사이의 시간 지연이 필요한 경우 사용한다.

- void SetWait(int time)

Input parameter

Int time: 1/1000 초 단위의 시간을 입력한다. 1 초 : 1000, 0.1 초 : 100

Result

없음.

사용예제

```
SetWait(1000); //1 초 delay
```

3. SendToLog

SendToLog 창에 로그를 표시하는 함수

- void SendToLog(CString str, CString color)

Input parameter

CString str : 로그창에 표시할 정보를 입력한다.

CString color : 입력한 값을 보여줄 색을 결정한다

값이 없는 경우 black 으로 간주된다.

제공되는 컬러 : Green, Red, Blue, Black (대소문자는 가리지 않음)

Result

없음.

사용예제

```
SendToLog("Test..");
SendToLog("PASS", "blue");
SendToLog("FAIL", "red");
```

4. ToolQuit

Test Plan 을 진행 중에 스크립트 내에서 plan 실행을 중지하고자 할 때 사용

- void ToolQuit()

Input parameter

없음

Result

없음.

사용예제

```
If( FAIL ) ToolQuit(); //1 초 delay
```

5. SrvSetLogMode (*추가됨)

로그 레벨을 조절할 수 있는 함수

- BOOL SrvSetLogMode(int LogLevel)

Input parameter

int LogLevel : 0 부터 2 까지 로그레벨을 선택할 수 있다.

0 은 모든 로그, 1 은 correct data 만 보고 싶을 때, 2 는 correct data 와 MIC 에러를 보여준다.

Result

True : 설정 성공

False : 설정 실패

사용예제

SrvSetLogMode(0); // 0 : all, 1: correct data only, 2 : correct data + MIC error

6. SrvSetRFLogMode (*추가됨)

RF 값들을 로그에서 보여주는 함수

- BOOL SrvSetRFLogMode(int LogLevel)

Input parameter

int LogLevel : 0 부터 1 까지 로그레벨을 선택할 수 있다.

0 은 off,

1 은 Rx 패킷 및 Tx 패킷 raw data 를 보고자 하는 경우 설정한다.

Result

True : 설정 성공

False : 설정 실패

사용예제

SrvSetRFLogMode(1); // 0 : LoRa Raw data log off, 1 : On