



**CONTENTS DESCRIPTION:**-----

- rCOMM** -> OK. - communication well  
ERR. - communication failure
- rAIRNOF** -> ON. - air condition is turn on  
OFF. - air condition is turn off  
NONE - air contidion indicate communication failure so it can not returns any of previous status
- rAIRMODE** -> AUTO. - air condition is in auto mode  
COOL. - air condition is in cooling mode  
DRY. - air condition is in dehumidification mode  
HEAT. - air condition is in heating mode  
ONLYFUN. - air condition is in air-circulation only mode  
NONE. - klimanet can not returns any of previous status (communication failure or air condition is turn off)
- rAIRTEMP** -> INTEGER. - indicate temperature which is set up on air condition (value should move between 10 and 32)  
NONE. - klimanet can not returns any of previous status (communication failure or air condition is turn off)
- rAIRFUN** -> F1. - \  
F2. - \\  
F3. ---- one of five status of blower  
F4. - //  
F5. - /  
FA. - air condition is in auto mode
- rSWING** -> UD. - swing UP-DOWN is active (  
OFF. - swing is turn off  
NONE. - klimanet can not returns any of previous status (communication failure or air condition is turn off)
- rROOMT** -> STRING. - string, for ex. +27,5. (tenths are separate with comma)  
NONE. - communication failure, can not read value from air condition
- rTIMER** -> OFF/OFF. - on-timer deactive, off-timer deactive  
ON/OFF. - on-timer active, off-timer deactive  
OFF/ON. - on-timer deactive, off-timer active  
ON/ON. - on-timer active, off-timer active  
NONE. - communication failure, can not read value from air condition
- rREMOTE** -> ENA. - control the air condition by remote (IR) control allowed  
DIS. - control the air condition by remote (IR) control not allowed  
NONE. - communication failure, can not read value from air condition

**rERROR** -> HEXA VALUE. - hexa value of failure code, for ex. (03)  
---. - air condition is not indicate the failure  
NONE. - communication failure, can not read value from air  
condition

**rCHGD1..5** -> X.X.X.X. - instead X there is value NOR./CHG.  
NOR. - modification on air condition is not running  
CHG- - modification on air condition is running

**rNAME.** -> STRING. -> name of klimanet, max. size 16-char.

**rOLDVAL** -> X1.X2.X3.X4.X5. - while instead is :  
X1-mode (vid. rAIRMODE)  
X2- temperature in auto mode (see rAIRTEMP)  
X3- temperature in heating mode (see rAIRTEMP)  
X4- temperature in cooling mode (see rAIRTEMP)  
X5- indicate condition of blower

**rUSER.** -> STRING. - name of signed in client, max. size 16-char.  
---. - none signed in client

**rGRPMD.** -> X1.X2.X3.X4.X5.X6.X7.X8. - while  
X1. - indicate mode of klimanet in group  
0. - SINGLE  
1. - MASTER  
2. - SLAVE  
X2. - indicate if the on/off-parameter is grupper or not  
0. - ungrouped  
1. - gruppued  
X3. - indicate if the mode-parameter is gruppued or not  
0. - ungrouped  
1. - gruppued  
X4. - indicate if the temperature-parameter is gruppued or not  
0. - ungrouped  
1. - gruppued  
X5. - indicate if the blower-parameter is gruppued or not  
0. - ungrouped  
1. - gruppued  
X6. - indicate if the swing-parameter is gruppued or not  
0. - ungrouped  
1. - gruppued  
X7. - indicate if the swap in group is turn on or not  
0. - OFF  
1. - ON  
X8. - indicate value of offset in swap  
- number btw. 1-10

!! WRITE !!!

Protocol of commands or change of parameters is progressed by HTTP-POST request.

Next method SendData( <url> ) in programm language Java, contains all possible parameters and their values (KEY-VALUE par), which are acceptable to klimanet. Their description was listed above.

```
private void SendData( String IporDNSStr ) {  
//*****  
  
        List<NameValuePair> ParamName_ParamValue = new  
ArrayList<NameValuePair>(); // List of POST KEY-VALUE pairs  
  
        if ( KlimanetVars.get( "rAIRNOF" ).equals( "ON" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiON", "On"));  
        else if ( KlimanetVars.get( "rAIRNOF" ).equals( "OFF" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiON", "Off"));  
        else  
            ParamName_ParamValue.add(new BasicNameValuePair("wiON", "Err"));  
  
        if ( KlimanetVars.get( "rAIRMODE" ).equals( "AUTO" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE", "Auto"));  
        else if ( KlimanetVars.get( "rAIRMODE" ).equals( "DRY" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE", "Dry"));  
        else if ( KlimanetVars.get( "rAIRMODE" ).equals( "COOL" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE", "Cool"));  
        else if ( KlimanetVars.get( "rAIRMODE" ).equals( "HEAT" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE", "Heat"));  
        else if ( KlimanetVars.get( "rAIRMODE" ).equals( "ONLYFAN" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE",  
"OnlyFun"));  
        else  
            ParamName_ParamValue.add(new BasicNameValuePair("wiMODE", "Err"));  
  
            ParamName_ParamValue.add(new BasicNameValuePair("wiTEMP",  
KlimanetVars.get( "rAIRTEMP" ) ));  
  
        if ( KlimanetVars.get( "rAIRFUN" ).equals( "FA" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "FAuto"));  
        else if ( KlimanetVars.get( "rAIRFUN" ).equals( "F1" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Fun1"));  
        else if ( KlimanetVars.get( "rAIRFUN" ).equals( "F2" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Fun2"));  
        else if ( KlimanetVars.get( "rAIRFUN" ).equals( "F3" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Fun3"));  
        else if ( KlimanetVars.get( "rAIRFUN" ).equals( "F4" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Fun4"));  
        else if ( KlimanetVars.get( "rAIRFUN" ).equals( "F5" ) )  
            ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Fun5"));
```

```

else
    ParamName_ParamValue.add(new BasicNameValuePair("wiFUN", "Err"));

if ( KlimanetVars.get( "rSWING" ).equals( "UD" ) )
    ParamName_ParamValue.add(new BasicNameValuePair("wiSWNG", "Ud"));
else if ( KlimanetVars.get( "rSWING" ).equals( "OFF" ) )
    ParamName_ParamValue.add(new BasicNameValuePair("wiSWNG", "Off"));
else ParamName_ParamValue.add(new BasicNameValuePair("wiSWNG", "Err"));

postData( "http://" + IporDNSStr, ParamName_ParamValue );

}
//*****
*****
private HttpResponse postData( String target_url, List<NameValuePair> nameValuePairs )
{
    HttpResponse response = null;
    // Create a new HttpClient and Post Header
    HttpClient httpclient = new DefaultHttpClient();
    HttpPost httppost = new HttpPost( target_url );
    try {
        httppost.setEntity( new UrlEncodedFormEntity( nameValuePairs ) );
        // Execute HTTP Post Request
        response = httpclient.execute( httppost );
    } // try
    catch ( ClientProtocolException e ) {
        Log.d("Klimanet", "Error, HTTP-POST throws ClientProtocolException !!!" );
    } // catch
    catch ( IOException e ) {
        Log.d("Klimanet", "Error, HTTP-POST throws IOException !!!" );
    } // catch
    return response;
}
//*****

```