

Freelux

Fx_Plug command line description

V1.0.3 (16-09-26)

1. Base CMD Service:

1.1. Device looking: (search device in LAN network)

- Description: Listen the port 1607 as well as **udp broadcast**.
- Cmd: "Are You Freelux IOT Device?"
- Respond example:
`{"cmd":"looking","dType":"PLUG","mac":"00:00:00:c0:14:7b","ip":"238.158.33.64","uuid":"DVES_00C41D1A","fName":"Fx_Plug"}`

1.2. Find device by IP:

- Description: Find device when you know IP address.
- Cmd: "findD?"
- Respond example: "I'm Plug. 5c:cf:7f:12:2b:81 192.168.1.105 DVES_00C01293"

1.3. Get friend name command:

- Cmd: "fName?"
- Respond ex: `{"cmd":"fName?","fname":"Fx_Plug"}`

1.4. Set friend name command:

- Cmd: "fName={Name}"
- Respond ex: `{"cmd":"fName?","fname":"Fx_Plug"}`

1.5. Get hardware, firmware, OTA version:

- Cmd: "fbhVer?"
- Respond ex:
`{"cmd":"fbhVer?","firmware":" 10006","hardware":" 10000","boot":" 10000"}`

1.6. Reset device command:

- Cmd: "Rst=1"
- Respond ex: `{"cmd":"Rst=","status":"ok"}`

1.7. Get Host information command:

- Cmd: "HostInfor?"
- Respond ex: `{"cmd":"HostInfor?","addr":"192.168.1.124","port":1883}`

1.8. Set Host information command:

- Cmd: "HostInfor={hostname,host_port}"
- Respond ex: `{"cmd":"HostInfor=","addr":"hostname","port":port}`

1.9. Get user information command:

- Cmd: "UserInfor?"
- Respond ex: {"cmd":"UserInfor?","name":"DVES_USER","pass":"DVES_PASS"}

1.10. Set user information command:

- Cmd: "UserInfor={name,pass}"
- Description: Maximum length of name and pass is 70 byte.
- Respond ex: {"cmd":"UserInfor=","name":"Mr Linh","pass":"1233545464"}

1.11. Get RSI command:

- Cmd: "rsi?"
- Respond ex: {"cmd":"Rsi?","status":-51}

1.12. Get Local IP:

- Cmd: "lip?"
- Respond ex: {"cmd":"lip?","status":"192.168.1.111"}

1.13. Security set status (choosing MQTTs or MQTT):

- Cmd: "secur={value}"
- discript: MQTT (value = 0), MQTTs (value =1)
- Respond ex: {"cmd":"secur=","value": real value}

1.14. Security get status (choosing MQTTs or MQTT):

- Cmd: "secur?"
- Respond ex: {"cmd":"secur=","value": real value}

1.15. Custommer memory set:

- Cmd: "cusMem=custommer data"
- Description: Save a custommer data to Flash Memory
- CMD Ex:
 - CMD: cusMem={"setting1":"on"}
 - Rspnd: {"cmd":"cusMem=","value": "strFormat", "setting1":"on"}

1.16. Custommer memory get (>=firmware v1.0.9):

- Cmd: "cusMem=custommer data"
- Description: Load custommer data from Flash Memory
- Respond ex: {"cmd":"cusMem=","value": "custommer data"}

1.17. Time Zone get (>=firmware v1.0.9):

- Cmd: "timeZone?"
- Respond ex: {"cmd":"timeZone?","value": 7}

1.18. Time Zone set:

- Cmd: "timeZone={time zone value}"
- Respond ex: {"cmd":"timeZone?","value": 7}

1.19. Keep update information period set:

- Cmd: "keepUp={period value}"
- Unit: second
- Respond command ex: {"cmd":"keepUpd=","value": 60}

- Description: The plug will autor update UIPE after period time. If period time value is 0, it will disble this feature.
- Keep update message ex:
{"cmd":"uipe@", "status":"on", "u":2306, "i":0, "p":0, "e":57496}

1.20. Keep update information period get:

- Cmd: "keepUp?"
- Respond ex: {"cmd":"keepUpd=", "value": 60}

1.21. Calendar Refresh:

- Cmd: "caleResh="
- Description: Reload a Flash calendar to ram and include them.
- Respond ex:

1.22.

2. Config Wifi Network:

2.1. Set SSI and Password wifi for Plug by Smart Config:

2.2. Set SSI and Password wifi for Plug by manual config:

- Cmd: "ssid={Wifi_name,Wifi_pass}"
- Description: use this function when can't smart config.
- How to config:
 - Step 1: long long press button until you see fast blinking led .
 - Step 2: looking in your system



- Step 3: connect to FxDevice with pass-work: 12345678
- Step 4: Connect to IP :192.168.4.1 and port 1607 as TCP service
- Step 5: Send this command “ssid={Wifi_name,Wifi_pass}”
- Step 6: Reset Plug
- Respond ex: {"cmd": "ssid=", "name": "freeluxvn", "pass": "0915992468"}

3. MQTT & MQTTS:

Note :

- please set security value to change MQTT and MQTTS
- fx_device/global_topic/advertisement/device_id is global topic

3.1. MQTT:

- host name:
- host port: 1883
- user name:
- pass work:
- Mqtt control topic : fx_device/control/DVES_ID
- Mqtt report topic : kmDVES_ID

3.2. MQTTS:

- host name: m2m.eclipse.org
- host port: 8883
- user name:
- pass work:
- Mqtt control topic : fx_device/control/DVES_ID
- Mqtt report topic : fx_device/report/DVES_ID

3.3. SSL ca file:

3.3.1. CA Client:

Request minimum size ram 35Kb for SSL with key 1024bit.

Reference: TLS_BiDirectVerif_Demo

3.3.2. CA Server:

- Request a **root** ca server file from server (TLS.ca_x509.pem or TLS.ca_x509.cer).
- Convert .pem file to .cer or .der (.der and .cer file is same format)
- Using make_cacert.py tool to convert to esp_ca_cert.bin
- writing esp_ca_cert.bin to Flash at sector 530.

4. Plug CMD Service:

4.1. On/Off:

- Cmd: “on” or “off”
- Description: Turn on/off Plug
- Cmd Ex:
 - Cmd: “off”
 - Respond: {"cmd":"off","status":"ok","req":"Cmd"}

4.2. Get U,I,P,E information:

- Cmd: "uipe?"
- Unit:
 - U: V
 - I: mA
 - P: W
 - E: W/h
- Respond ex: {"cmd":"uipe?","status":"off","u":0,"i":0,"p":0,"e":805}

4.3. Energy reset:

- Cmd: "RstE=1"
- Description: Reset energy value to zero.
- Respond ex: {"cmd":"RstE=","status":"ok"}

4.4. Get Calib U,I,P,E cmd:

Cmd: "CalibS?"

- Respond: {"cmd":"CalibS?","Ku": %d, "Ki": %d, "Kp": %d, "Ke":%d}\n"

4.5. Set Calib U,I,P,E cmd:

- Cmd: "CalibS={Ku_xxxx,Ki_xxxx,Kp_xxxx,Ke_xxxx,pas_11002233}"
- Respond: {"cmd":"CalibS=","Ku": %d, "Ki": %d, "Kp": %d, "Ke":%d}\n"
- Description:
 - The value calibration is interger value
 - $real_valueU15s = Ku15s * U15s$
 - $Ku = (real_valueU15s / U_power_meter)$.
 - $Ki = (real_valueI15s / I_power_meter)$.

- $Kp = (\text{real_valueP15s} / P_power_meter).$
- $Ke = (\text{real_valueE15s} * E_power_meter);$

4.6. Autor Calib U,I,P,E cmd:

Note: Please don't use this funtion if you are not sure.

- Cmd: "CalibA=",p
- Respond:

4.7. Plug parametter set:

- Cmd:
"param={uMax_value,iMax_value,pMax_value,uMin_value,iStart_value,tStart_value}"
- Description:
 - Will be to protect device.
 - uMax,iMax,pMax: voltage, current, power maximum.
 - iStart: Starting current
 - tStart: time for starting current. (unit: milli second)
- Note:
 - U,I,P need to be multiplicat 10.
- Ex command:
param={uMax_2600,uMin_1980,iMax_70000,iStart_160000,tStart_400,pMax_36800}

4.8. Plug parametter Get:

- Cmd: "param?"
- Respond:
{ "cmd": "param?", "uMax": 2600, "iMax": 70000, "pMax": 36800, "uMin": 1980, "hbTime": 2053543, "iStart": 160000, "tStart": 400 }

4.9. Fail report Get:

- Cmd: "failSt"
- Respond:
- Description: get a fail status ID and clear them.
- Fail ID mask:

○ OVER_VOLTAGE	0x0001
○ OVER_CURRENT	0x0002
○ OVER_LOAD	0x0004
○ BROWN_OUT_VOLTAGE	0x0008
○ NO_VOLTAGE_STRIP	0x0010
○ RELAY_SHORTED	0x0020

5. Wifi Bridge RF433 Service (day 2016_14_12):

5.1. Add new Rf device to base:

- Cmd: "rfAddDev={fName}"
- Description: add a new device with a friendly name.
- Cmd Ex:
 - Cmd: "rfAddDev="
 - Respond: { "cmd": "rfAddDev=", "id": 586229768 }

5.2. Remove RF device:

- Cmd: "rfRmv={id}"
- Description: Remove a device in bridge. It will return number exist device
- Cmd Ex:
 - Cmd: "rfRmv={2028865523}"
 - Respond: {"cmd":"rfRmv=","count":0}

5.3. Get list of device in bridge by index:

- Cmd: "rfDevLisIx?{startIndex,endIndex}"
- Note: endIndex-startIndex <25 end endIndex < Number Device
- Description: Get list of device
- Cmd Ex:
 - Cmd: devLisI?{0,1}
 - Respond:
{"cmd":"rfDevLisIx?","ixStart":0,"ID":[{"id":949427414,"fName":"sofar"}, {"id":281638437,"fName":"hotel"}]}

5.4. Rf get device number:

- Cmd: "rfGetDevN?"
- Description: Request a device number which is managed by Bridge
- Cmd Ex:
 - Cmd on: rfGetDevN?
 - Respond: {"cmd":"rfGetDevN?","count":8}

5.5. Rf set status:

- Cmd: rfSetSt={id,on} or rfSetSt={id,off}
- Description: Set status of Rf Device
- Cmd Ex:
 - Cmd on: rfSetSt={1813771804,on}
 - Respond: {"cmd":"rfSetSt=","on":"ok"}

5.6. Rf change friendly name:

- Cmd: rfCN={ID,fName}
- Description: Change friendly name of device
- Cmd Ex:
 - Cmd on: rfCN={1338990198,linh243}
 - Respond: {"cmd":"rfCN=","id":1338990198,"newName":"linh243"}

//Quoc added

5.7. Remove all RF device

- Cmd: "rfRmvA"
- Description: Remove all FRX103 plugs in the repeater. It will return number of available FRX103 plugs.

- Cmd Ex:
 - Cmd: "rfRmvA"
 - Respond: {"cmd":"rfRmvA=","status":"ok"}

5.8. Add icon

- Cmd: "rfRmvA"
- Description: Remove all FRX103 plugs in the repeater. It will return number of available FRX103 plugs.
- Cmd Ex:
 - Cmd: "rfRmvA"
 - Respond: {"cmd":"rfRmvA=","status":"ok"}

6. Dimmer CMD Service:

6.1. Set Dim value

- Cmd: dim={value}
- Description: Set dim value for FRX104 WiFi Dimmer.
- Cmd Ex:
 - Cmd: dim={20}
 - Response: {"cmd":"dim=","value":20}

6.2. Get Dim value

- Cmd: dim?
- Description: Get dim value from FRX104 WiFi Dimmer..
- Cmd Ex:
 - Cmd: dim?
 - Response: {"cmd":"dim?","status":"on","value":20}

6.3. Set Profile :

- Cmd: profile={value}
- Description: Set profile of the dimmer
- Cmd Ex:
 - Cmd: profile={20}
 - Response: {"cmd":"profile=","value":20}

6.4. Get Dim Profile

- Cmd: profile?
- Description: Get profile of the dimmer
- Cmd Ex:
 - Cmd: profile?
 - Response: {"cmd":"profile?","value":20}
 -

7. Wall Switch Touch:

7.1. Find device:

- Description: Find device.
- Cmd: "findD?"
- Respond example:

```
{ "cmd": "looking", "dType": "WALL_SWITCH_TOUCH", "mac": "00:00:00:00:00:00", "ip": "0.0.0.0", "uuid": "DVES_00C10491", "fName": "Fx_Wall_Switch_Touch", "iType": 0, "caAct": 101, "subDev": { "numSDVE": 3, "ID0": { "dType": "WST", "fName": "TOUCH_SWITCH1" }, "ID1": { "dType": "WST", "fName": "TOUCH_SWITCH2" }, "ID2": { "dType": "WST", "fName": "TOUCH_SWITCH3" } } }
```

7.2. On/Off Switch

- Cmd: on={id} or off={id}
- Description: set on/off switch touch
- Cmd Ex:
 - Cmd: on={0}
 - Response: { "cmd": "on0", "status": "ok", "req": "Cmd" }

7.3. Get name of sub device:

- Cmd: subFName?{ch}
- Description: the chanel is 0 to 2
- Cmd Ex:
 - subFName?{1}
 - Response: { "cmd": "subFName=", "ID": 1, "subFName": "name of sub device" }

7.4. Change name of sub device:

- Cmd: subFName={ch,"name of sub device"}
- Description: the chanel is 0 to 2
- Cmd Ex:
 - subFName={1,"name of sub device"}
 - Response: { "cmd": "subFName=", "sFname": "name of sub device", "ID": 1 }

7.5. Set number chanel touch:

- Cmd: subDN={value}
- Description: the max number chanel is 3
- Cmd Ex:
 - Cmd: subDN={2}
 - Response: { "cmd": "subDN=", "value": 2 }
 -

