

# XIAOME Vacuum cleaner Robot

## Nutzung der WLAN-Schnittstelle zur Steuerung per Python

### Ermittlung des TOKEN

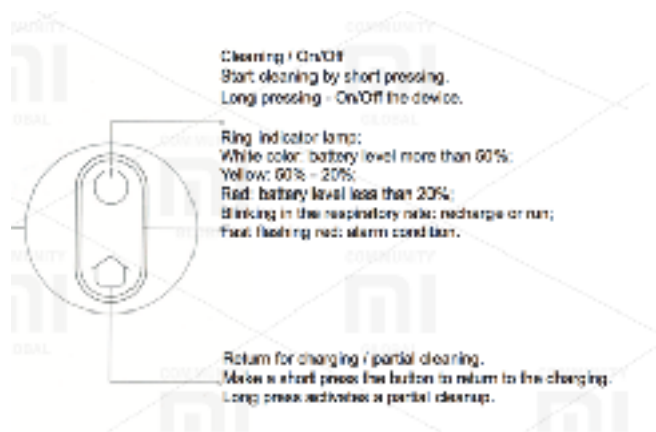
Der *Token* ist eine eindeutige Identifizierung des Roboters und wird automatisch neu erzeugt sobald der Roboter in das Heimnetzwerk eingebunden wird.

Benötigte Hilfsmittel zur Ermittlung des *Tokens*

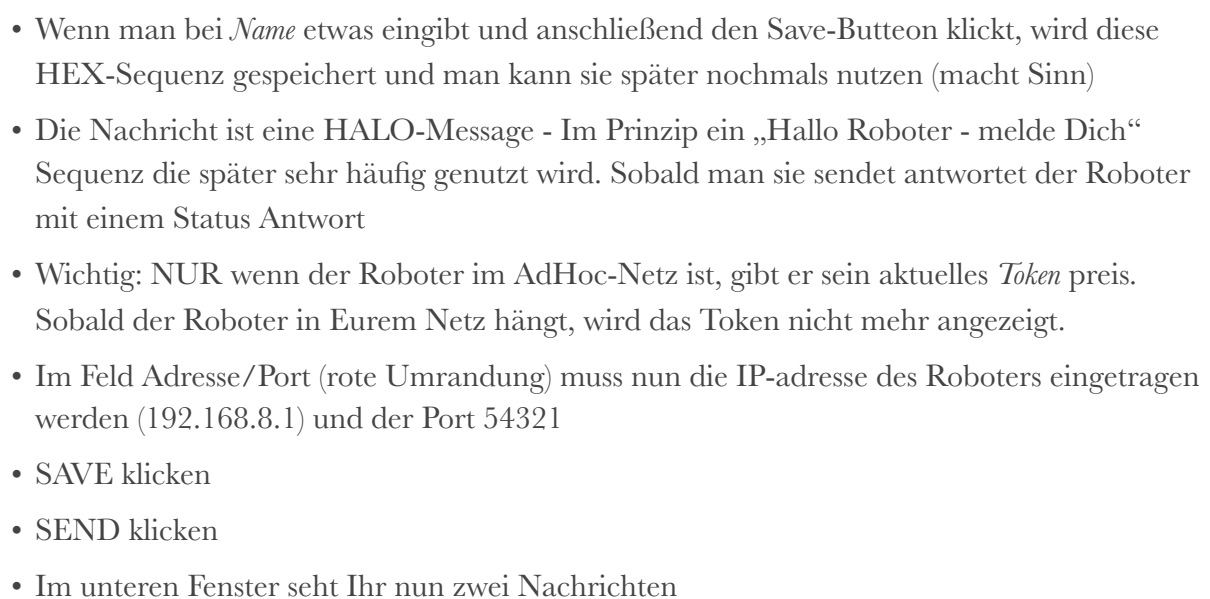
- *Python 2.7 installiert*
- *Wireshark installiert* (<https://www.wireshark.org/>)
- *Paket-Sender installiert* (<https://packetsender.com/>)

Vorgehensweise:

- Die WLAN Verbindung des Roboters zurücksetzen indem man sowohl POWER als auch HOME gleichzeitig für ca 5-8sek drückt. Der Roboter piept dann irgendwann mal und die blaue LED fängt an zu blinken
- Wenn die App auf dem Smartphone installiert ist, das Device aufrufen und in den Einstellungen das Device entfernen. Sollten ihr die chinesische Einstellungen noch haben, dann müsst ihr etwas suchen :-O
- Wenn das WLAN des Roboters zurückgesetzt wurde und das Device in der App entfernt wurde, dann sollte anschließend in Eurem Netzwerk eine neue SSID erscheinen „Rockroboxxxx“. Diese auswählen und Euren Rechner mit dem Adhoc Netz verbinden
- In der Regel wird die IP Adresse 192.168.8.1 für den Roboter ausgewählt. Ihr könnt das sicherheitshalber prüfen indem ihr auf der Konsole die IP-Adresse Eures Computers überprüft. Der Roboter hat im gleichen Netz nun die .1 Adresse
- Möglichkeit 1 zur Ermittlung des Token
  - Paketsender starten



- „21310020#####“



- die zweite ist EURE gesendete Nachricht (FROM = YOU) TO=Roboter
- und die Antwort des Roboters. Diese fängt immer mit 213100 an
- Ermittlung des *Token*: Das sind jetzt die LETZEN 16 Bytes (im Bild als FF gekennzeichnet)
- WICHTIG: Diese HEX-Werte kopieren und speichert die Euch in eine Text-Datei.
- **Das ist Eurer Token, der NUR auf diesem Roboter und NUR solange funktioniert, wie der Roboter in Euerem Netz eingebunden wird. Sobald Ihr den Roboter in ein neues Netz einbinden, muss der Token neu ermittelt werden !!!!**
- Jetzt könnt Ihr den Roboter ganz normal in Euer Netz einbinden (über die App). Der Token kann nun zukünftig über das python Skript verwendet werden

# Nutzung des Python Skripts

Das Skript wurde unter Python 2.7 entwickelt und getestet ob es auch unter 3.5 läuft kann ich derzeit nicht sagen, da ich es nicht getestet habe.

## Vorbereitung:

- Da die Verbindung zwischen dem Computer und dem Roboter verschlüsselt wird muss die Python cryptography Bibliothek installiert werden

- Voraussetzung ihr habt *pip* installiert.

```
pip install cryptography
```

- Bitte beachten: Wenn ihr neben 2.7 auch 3.5 installiert habt, dann ist häufig der pip Befehl mit 3.5 verbunden. Spricht Installationen werden für 3.5 durchgeführt. Prüfen könnt ihr das mit `pip -V`. Anschließend bekommt ihr eine Anzeige wo der pip Befehl liegt. Entweder im Bereich 2.7 oder 3.5

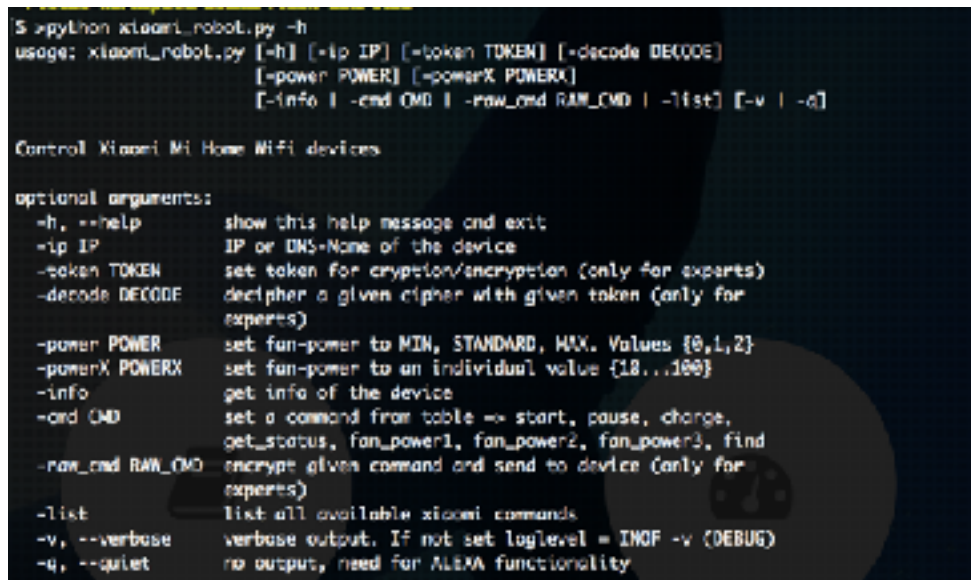
- das Python Skript in Eurem Home-Verzeichnis installieren (kopieren) und die Ausführrechte vergeben (unter MacOS/Unix)

```
chmod a+x xiaomi_roboter.py
```

- Nun das Skript wie folgt starten:

```
python xiaomi_robot.py -h
```

- Wenn alles sauber funktioniert muss nachfolgende Anzeige kommen



```
S>python xiaomi_robot.py -h
usage: xiaomi_robot.py [-h] [-ip IP] [-token TOKEN] [-decode DECODE]
                    [-power POWER] [-powerX POWERX]
                    [-info I] [-cmd CMD] [-raw_cmd RAW_CMD] [-list] [-v] [-q]

Control Xiaomi Mi Home Wifi devices

optional arguments:
  -h, --help            show this help message and exit
  -ip IP                IP or DNS-Name of the device
  -token TOKEN          set token for crypton/encryption (only for experts)
  -decode DECODE        decipher a given cipher with given token (only for
                        experts)
  -power POWER          set fan-power to MIN, STANDARD, MAX. Values {0,1,2}
  -powerX POWERX        set fan-power to an individual value {18...100}
  -info                 get info of the device
  -cmd CMD              set a command from table => start, pause, change,
                        get_status, fan_power1, fan_power2, fan_power3, find
  -raw_cmd RAW_CMD      encrypt given command and send to device (only for
                        experts)
  -list                 list all available xiaomi commands
  -v, --verbose          verbose output. If not set loglevel = INFO -v (DEBUG)
  -q, --quiet            no output, need for ALIEN functionality
```

- Kommen Fehlermeldungen müssen diese vorher behoben werden. In der Regel liegt das an einer fehlenden Bibliothek die dann über *pip install* nachinstalliert werden muss. Da ich aber Eure Rechner nicht kenne, kann ich das leider nicht genau beschreiben
- Das Skript kann prinzipiell auch für andere Xiaomi-Devices genutzt werden, mangels Testgeräte konnte ich dies aber leider bis dato nicht testen

- Nun kommt das *Token* in Spiel.
- **Test 1**
  - Folgende Befehlssequenz eingeben:

```
~/local-workspace/alex/Flask-esk/Flask
$ python xiaomi_robot.py -ip "192.168.1.100" -cmd "status" -token "31684300000000000000000000000000"
INFO:xiaomi:*****
INFO:xiaomi:Device Type      : Xiaomi Mi Robot Vacuum      TD(834c)
INFO:xiaomi:IP-Address       : ID(192.168.1.100)
INFO:xiaomi:Token            : backend                     TD(31684300000000000000000000000000)
INFO:xiaomi:encoded token    : ID(1hCk00000000000000000000000000000000)
INFO:xiaomi:*****
INFO:xiaomi:SendReq Xiaomi to Host :
Data:({ "result": [ { "msg_ver": 4, "msg_seq": 834, "state": 0, "battery": 100, "clean_time": 0, "clean_area": 0, "error_code": 0, "msg_present": 1, "in_cleaning": 0, "fan_power": 60, "dnd_unchilled": 1 } ], "id": 1000 })
```

- Als -ip gebt ihr bitte die IP-Adresse EURES Roboters ein
- Als -token gebt ihr bitte EUREN Token ein.
- Alls Strings müssen mit Anführungszeichen eingegeben werden
- Als -cmd versuchen wir den Status des Roboters abzufragen
- Wenn alles klappt, antwortet der Roboter mit dem aktuellen Status (unterer Rand)
- Hier sieht man z.B. den Akkustand (battery) = 100%, fan-power: 60 (Standard)
- Seht ihr das dann habt ihr gewonnen !
- **Test 2**
  - statt -cmd „status“ nun -cmd „start“ eingeben => Der Roboter fährt los
  - -cmd „stop“ oder -cmd „pause“ => pausiert den Roboter
  - -cmd „charge“ oder -cmd „home“ => Roboter fährt zur Ladestation
- Test 3
  - nun könnt ihr etwas mit den Parametern spielen. Die unter Test 2 sind aber die wichtigen.
  - Übrigens es gibt auch einen Trick um die Remote-Control-Funktion über das Python-Skript auszuführen - ist aber mühselig und aus meiner Sicht so ggf. nicht nutzbar.

## Fazit:

Grundsätzlich funktioniert das Ganze und man kann dieses Skript ins seine Haussteuerung einbauen (z.B. FHEM) oder aber über einen Skill mit Alexa verbinden. Diesen Skill habe ich in einer Alpha-Version schon fertig gestellt es gibt aber noch Bugs

Viel Spass

LunaX ([lunax42@gmx.de](mailto:lunax42@gmx.de))

