XIAOMI Vacuum cleaner Robot

Nutzung der WLAN-Schnittstelle zur Steuerung per Python

Version 0.2

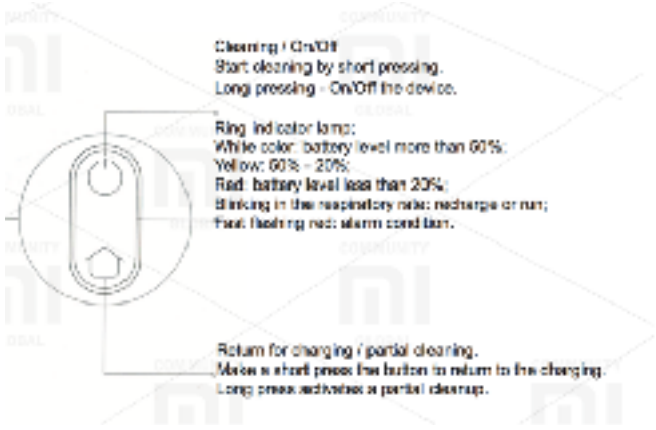
Ermittlung des TOKEN

Der *Token* ist eine eindeutige Identifizierung des Roboters und wird automatisch neu erzeugt sobald der Roboter in das Heimnetzwerk eingebunden bzw. neu 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. Solltet 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 „*Rockroboxxxxx*“. Diese AdHoc-Netz auswählen und Euren Rechner mit diesem Netz verbinden.
- In der Regel wird die IP Adresse 192.168.8.1 für den Roboter konfiguriert. Ihr könnt das sicherheitshalber prüfen indem ihr auf der Konsole die IP-Adresse Eures Computers überprüft. Der Roboter hat im gleichen Netz die xxx.yyy.zzz.1 Adresse

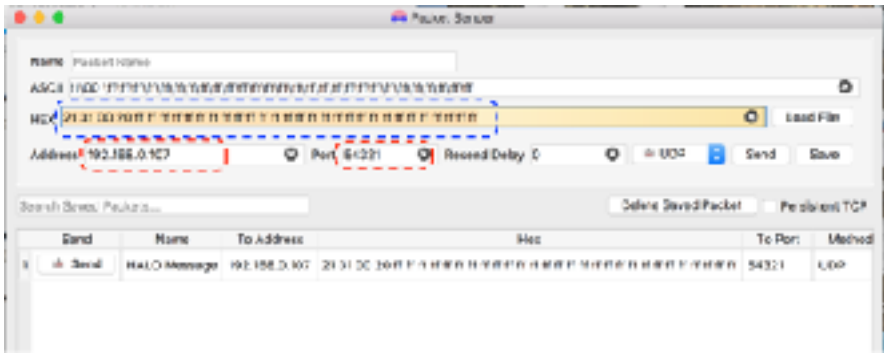
Möglichkeit 1 zur Ermittlung des *Token*

- via dem Python-Skript
- Zuerst muss das Python-Skipt installiert werden (siehe „Nutzung des Python-Skripts)
- Sobald das Skript fehlerfrei funktioniert folgende Eingabe machen
  - python xiaomi_robot.py -info
  - Nach einigen Sekunden sollte eine Antwort vom Roboter zurück kommen. Mit der Angabe des Tokens. Das Token muss 16 Bytes umfassen, ansonsten ist etwas falsch gelaufen. In diesem Falle wiederholen oder mit Möglichkeit 2 probieren

Möglichkeit 2 zur Ermittlung des *Token*

- Paketsender starten
- im Feld HEX folgende Werte eintragen (ohne Anführungszeichen) (blau umrandetes Feld)

„21310020ffffffffffffffffffffffffffffffffffffffffffffffffffffffff“



- Wenn man bei *Name* etwas eingibt und anschließend den Save-Butteon klickt, wird diese HEX-Sequenz gespeichert und man kann sie später nochmals nutzen (macht Sinn)
- Die Nachricht ist eine HALO-Message - Im Prinzip ein „Hallo Roboter - melde Dich“ Sequenz die später sehr häufig genutzt wird. Sobald man sie sendet antwortet der Roboter mit einem Status Antwort
- **Wichtig**: NUR wenn der Roboter im AdHoc-Netz ist, gibt er sein aktuelles *Token* preis. Sobald der Roboter in Eurem Netz hängt, wird das Token nicht mehr angezeigt. Statt dass wird immer nur „FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF“ angezeigt
- Im Feld Adresse/Port (rote Umrandung) **muss** nun die IP-adresse des Roboters eingetragen werden (z.B. 192.168.8.1) und der Port 54321
- SAVE klicken
- SEND klicken
- Im unteren Fenster seht Ihr nun zwei Nachrichten

- die zweite ist EURE gesendete Nachricht (FROM = YOU) TO=Roboter
- und darüber 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)
  - Beispiel: 3168436B20AF52F0724F723033664E6F
- WICHTIG: Diese HEX-Werte kopieren und speichert die Euch in eine Text-Datei. Sie wird anschließend für das Python-Skript verwendet
- **Das ist EURER *Token*, der NUR auf diesem Roboter und NUR solange funktioniert, wie der Roboter in EUREM Netz eingebunden ist. Sobald Ihr den Roboter in ein neues Netz einbinden, muss der Token neu ermittelt werden !!!!. Auch bei einem erneuten Reset des WLAN wird ein neuer *Token* generiert**
- 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.

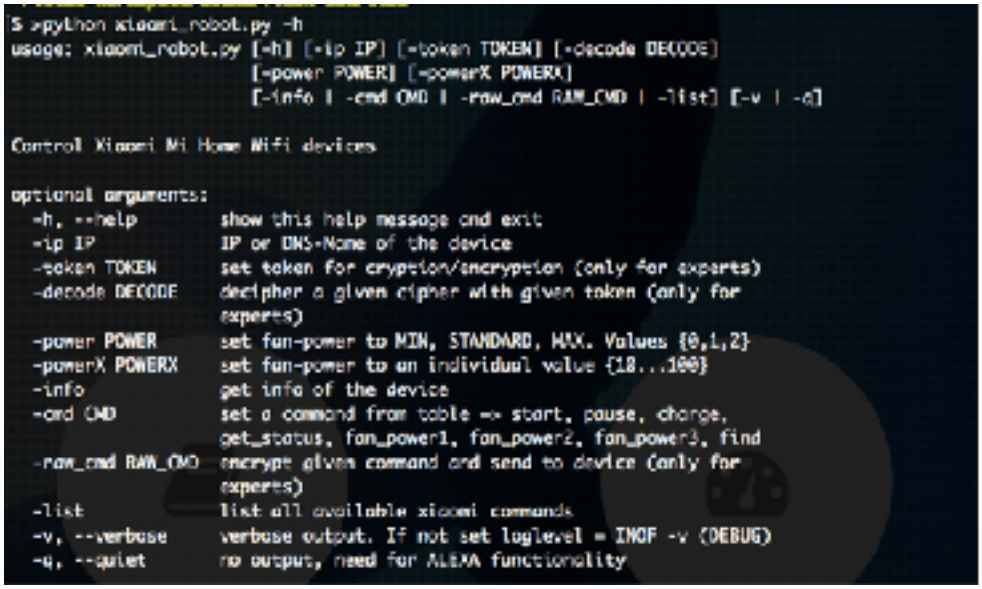
Vorbereitung:

- 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“. Das heißt 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. Sollte er auf 3.5 sich beziehen, dann müsst ihr den pip Befehle für 2.7 suchen oder nach installieren.
  - das Python Skript in Eurem Home-Verzeichnis installieren (kopieren) und die Ausführrechte vergeben (unter MacOS/Unix)

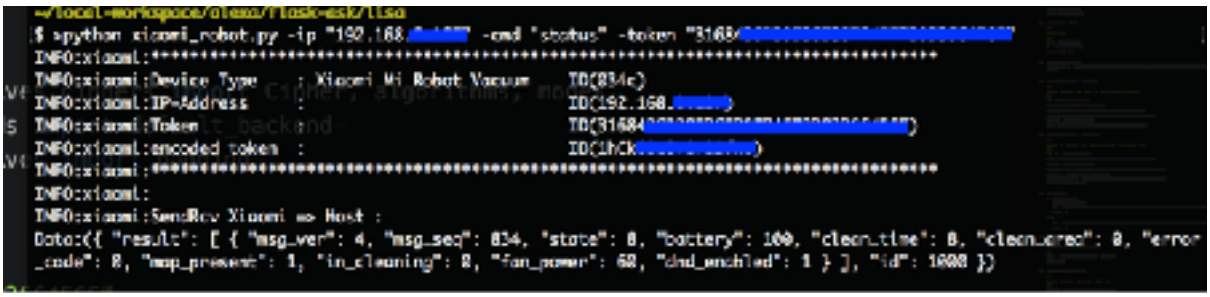
    ```
    chmod a+x xiaomi_roboter.py)
    ```
  - Nun das Skript wie folgt starten:

    ```
    python xiaomi_robot.py -h
    ```
  - Wenn alles sauber funktioniert muss nachfolgende Anzeige kommen
  - Kommen Fehlermeldungen müssen diese vorher behoben werden. In der Regel liegt das an einer fehlenden Bibliothek, die dann über `pip install <paket>` 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:



  - 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 senden wir ein Kommende an den Roboter. Im Beispiel lesen wir den Status des Roboters.
  - 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.
  - mit -info oder -list bekommt ihr alle möglichen Roboter-Kommandos angezeigt.
  - 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. Derzeit aber auch noch nicht im Skript implementiert.

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 (berndklein42@gmail.com)