

| Plugin-Daten     |                                                                                                                                                                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Autor            | <a href="#">Michael Schlenstedt</a>                                                                                                                                                                                                         |
| Logo             |                                                                                                                                                            |
| Status           | STABLE                                                                                                                                                                                                                                      |
| Version          | 1.3.0.1                                                                                                                                                                                                                                     |
| Min. LB Version  | 1.4.1                                                                                                                                                                                                                                       |
| Release Download | <a href="https://github.com/mschlenstedt/LoxBerry-Plugin-MiRobot2Lox-NG/archive/refs/tags/mirobot2lox-ng-1.3.0.1.zip">https://github.com/mschlenstedt/LoxBerry-Plugin-MiRobot2Lox-NG/archive/refs/tags/mirobot2lox-ng-1.3.0.1.zip</a>       |
| Beschreibung     | Das MiRobot2Lox-NG Plugin ermöglicht die Kommunikation zwischen bis zu 5 Mi Vacuum Robots und dem Loxone MiniServer.                                                                                                                        |
| Sprachen         | EN, DE, CZ                                                                                                                                                                                                                                  |
| Diskussion       | <a href="https://www.loxforum.com/forum/projektforen/loxberry/plugins/168084-plugin-mirobot2lox-ng-xiaomi-vacuum-robots">https://www.loxforum.com/forum/projektforen/loxberry/plugins/168084-plugin-mirobot2lox-ng-xiaomi-vacuum-robots</a> |

# MiRobot2Lox-NG

[Version History...](#)

[Änderungshistorie](#)

## Ursprung des Plugins



Dieses Plugin basiert auf dem Original-Plugin von [Dieter Schmidberger](#). Leider wird das Original-Plugin nicht mehr weiterentwickelt. Da es noch nicht auf LoxBerry V1.0 angepasst war und mittlerweile auch bei der Installation der notwendigen Python-Tools Probleme bereitet, wurde diese Version ins Leben gerufen. Die grundlegenden Ideen und das Konzept wurden vom Originalplugin übernommen - dennoch ist es von Grund auf neu programmiert und nutzt eine andere Kommunikationsschnittstelle zum Miniserver. Durch die direkte Integration in LoxBerry V1.0 konnten viele Standardfunktionen aus LB1.0 übernommen werden.

## Aufgabe des Plugins

Mit dem Plugin können Saugroboter der Firma Xiaomi (<https://xiaomi-mi.com/cleaning-gear/>) und der Firma Roborock (<https://en.roborock.com/>) gesteuert werden. Es können sowohl Befehle an den Roboter abgesetzt werden als auch Daten vom Roboter ausgelesen werden (z. B. den aktuellen Status, diverse Verbrauchszähler, usw.). Alle Daten können in LoxConfig visualisiert werden und auch in Logiken verwendet werden. Die Ausgangsbefehle an den Roboter können ebenfalls aus der Logik in LoxConfig heraus abgesetzt werden.

# Download

- Das Plugin-Archiv (ZIP) kann auf GIT-Hub heruntergeladen werden: <https://github.com/mschlenstedt/LoxBerry-Plugin-MiRobot2Lox-NG/releases>
- Der Sourcecode ist auf GitHub verfügbar: <https://github.com/mschlenstedt/LoxBerry-Plugin-MiRobot2Lox-NG>

# Fehlerberichte

Fehlerberichte bitte direkt im [Repository auf GitHub](#) melden.

# Konfigurationsoptionen

## Einstellungen

### Hole Daten vom Robot

Sollen Statusmeldungen vom Robot abgeholt werden, muss diese Option eingeschaltet werden (das will man normalerweise ). Alle 60 Sekunden reicht normalerweise aus. Wer auf Grund einer Logik wirklich aktuellere Daten haben muss, kann auch noch auf 30 Sekunden umschalten. Eine kürzere Zeit gibt es nicht, da es doch recht lange dauert, bis der Robot antwortet. Wer mehr als 2 Robots im Einsatz hat, sollte die Zeit etwas hochnehmen, da in der gewählten Periode alle Robots nacheinander abgefragt werden.

### Sende per UDP

Wer die Statusdaten per UDP zum Miniserver senden und dort per Virtuellen Eingängen weiterverarbeiten möchte, muss diese Option aktivieren. Eine andere Alternative ist, die Daten per Virtuellen HTML Eingang abzufragen. Das verursacht aber mehr Netzwerkverkehr, da bei UDP die Daten nur übertragen werden, wenn sie sich geändert haben. Daher ist UDP die bessere Wahl. Nur bei Netzwerkproblemen ist häufig HTML die einfacherere Variante.

### UDP Port

Der Port, auf dem der Miniserver auf neue Daten lauscht.

### Miniserver

Wähle den Miniserver aus, an den die Daten per UDP gesendet werden sollen. Es kann nur ein Miniserver ausgewählt werden.

## **Sende per HTML**

"Senden" ist eigentlich nicht der richtige Ausdruck. Es wird eine Text-Internetseite zur Verfügung gestellt, die per Virtuellem HTML Eingang vom Miniserver abgerufen werden kann. Diese Option ist immer aktiv.

## **Robot #1 - #5**

### **Aktiviere Robot**

Mit dieser Option aktivierst Du den Robot. Aktivieren heisst, dass von diesem Robot die Statusdaten abgerufen werden.

### **Gerätetyp**

Das Plugin unterstützt zwei unterschiedliche Protokolle, um mit dem Robot zu kommunizieren. Wähle hier aus, welchen Robot Du hast.

### **IP-Adresse**

Die IP-Adresse des Robots. Diese darf sich nicht ändern. Konfiguriere Deinen Router entsprechend.

### **Token**

Der Netzwerkverkehr zum Robot ist per "Token" gesichert. Ohne den Token kann man mit dem Robot nicht kommunizieren. Leider kann der Token aus Sicherheitsgründen nicht automatisch aus dem Robot ausgelesen werden. Bitte folge dieser Anleitung, um den Token auszulesen: [Token extrahieren](#)

Der ausgelesene Token muss hier eingegeben werden.

### **Zeit Ladestation verlassen (s)**

Es gibt einen Befehl "Dock Release" bzw. "Ladestation verlassen", der an den Robot abgesetzt werden kann. Bei diesem Befehl fährt der Robot x Sekunden geradeaus nach vorne (z. B. unter einem Schrank hervor). Die Zeit kann hier eingestellt werden. Der Befehl kann z. B. genutzt werden, um den Robot zum Staubbehälter-Leeren vorzufahren.



**Hinweis**

Leider scheint die Funktion "Dock Release" nicht mehr vernünftig zu funktionieren. Wenn Ihr damit Probleme habt, dann lasst den Robo direkt zu definierten Koordinaten fahren. Das geht, indem man ihn in X-Richtung z. B. 1m vorfahren lässt. Siehe unten unter "Manuelle Koordinatenermittlung"

## Robot Kommandos

Hier findest Du eine Auflistung aller Kommandos, die für die Robots voreingerichtet sind. Die Kommandos werden per HTTP REST API an das Plugin übergeben. Du kannst so auch eigene Kommandos generieren. Zum Absetzen der Befehle an den Robot wird die Software **miocli** verwendet (Achtung! Die Syntax von miocli unterscheidet sich leicht von mirobo! mirobo-Befehle funktionieren nicht!) . Dokumentation

hier: <https://python-miio.readthedocs.io/en/latest/miio.html#module-miio.vacuum>

Um einen Befehl an den Robot abzusetzen, wird folgende API verwendet (die möglichen Befehle finden sich in der miio-Dokumentaion):

### Kommando ohne Option, z. B. "find":

<http://loxberry/plugins/mirobot2lox-ng/sendcmd.cgi?command=find&robot=1&debug=0> command= **Kommando, welches abgesetzt werden soll (siehe miio Dokumentation)** robot=1|2|3|4|5 **Nummer des Robots, an den das Kommando gesendet werden soll** **debug=0|1** Debugging-Informationen aus oder an

### Kommando mit Option, z. B. "set\_fan\_speed":

[http://loxberry/plugins/mirobot2lox-ng/sendcmd.cgi?command=set\\_fan\\_speed&option=50&robot=1&debug=0](http://loxberry/plugins/mirobot2lox-ng/sendcmd.cgi?command=set_fan_speed&option=50&robot=1&debug=0) command= **Kommando, welches abgesetzt werden soll (siehe miio Dokumentation)** option= **Zusätzliche Option für das Kommando (siehe miio Dokumentation)** robot=1|2|3|4|5 **Nummer des Robots, an den das Kommando gesendet werden soll** **debug=0|1** Debugging-Informationen aus oder an

## Soundpacks

Hier kannst Du neue Soundpacks auf den Robot installieren, z. B. Deutsche Sprache. Wir haben verschiedene TTS Engines verwendet. Die beste Sprachqualität liefert normalerweise Amazon Polly. Nachdem Du ein Soundpack ausgewählt hast, musst Du noch festlegen, auf welchen Robot Du das Soundpack spielen möchtest. Anschließend wird das entsprechende Kommando ausgeführt.

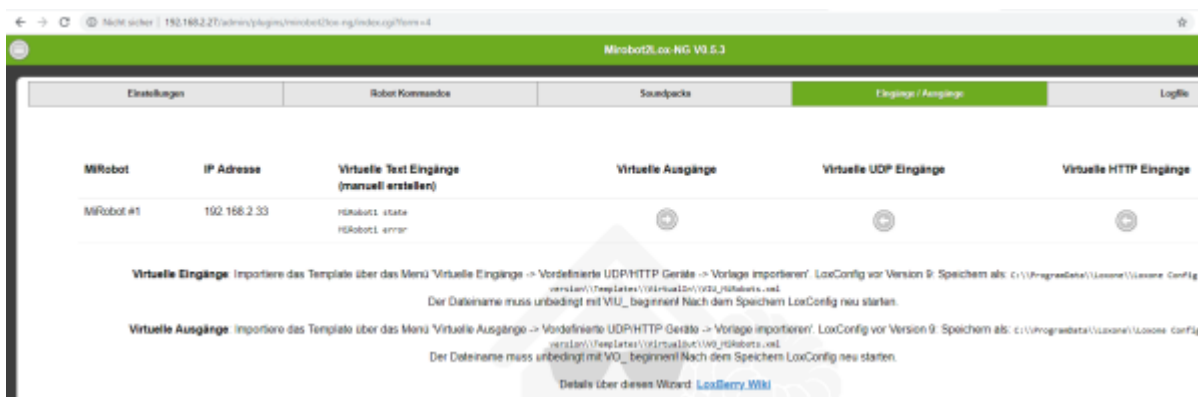


**Hinweis**

Der Robot muss über die XIAOMI App zuvor auf englische Sprache gestellt werden. Der Befehl zum Installieren des Soundpacks überschreibt die englischen Sprachdateien im Robot.

## Eingänge / Ausgänge

Mit diesem Wizard kannst Du sehr einfach die notwendigen Templates für die Ein- und Ausgänge in LoxConfig erstellen. Die Templates können dann sehr einfach ins eigene Projekt importiert werden und Du brauchst so nicht alle Ein- und Ausgänge manuell anzulegen. Nur Virtuelle Text Eingänge müssen manuell angelegt werden.

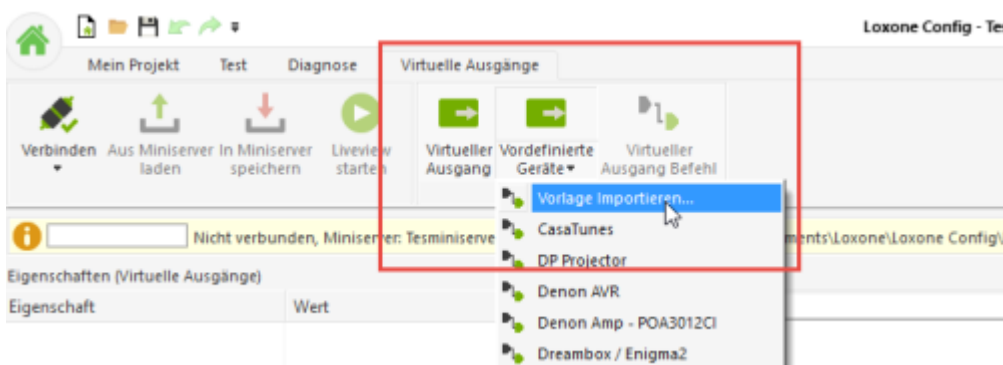


### Virtuelle Texteingänge

Die Virtuellen Texteingänge müssen manuell angelegt werden. Die Bezeichnung oder der Name müssen dem im Wizard angezeigten exakt entsprechen. Diese Virtuellen Textausgänge kann man dann z. B. auf einen Statusbaustein zur Anzeige in der App legen.

### Virtuelle Ausgänge

Ab LoxConfig Version 9 kannst Du die Datei direkt importieren:



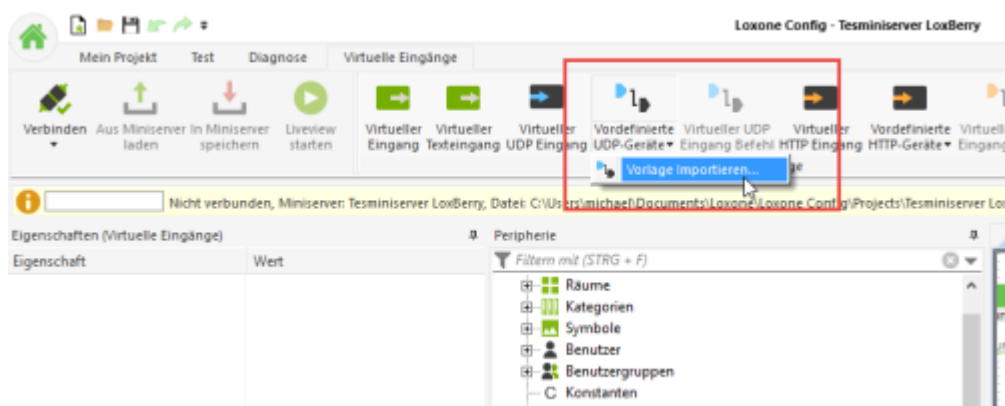
Bis LoxConfig Version 8 muss die Datei manuell in das

Verzeichnis C:\ProgramData\Loxone\Loxone Config

VERSION\Templates\VirtualOut kopiert werden. Sie muss dabei unbedingt mit **VO\_** beginnen, damit sie erkannt wird. LoxConfig muss anschließend neu gestartet werden.

## Virtuelle UDP Eingänge

Wenn Du die Status-Daten per UDP im Miniserver empfangen möchtest (Hinweis: Dazu muss die entsprechende Option in den Einstellungen des Plugins aktiviert sein), musst Du dieses Template in Dein Projekt integrieren. Ab LoxConfig Version 9 kannst Du die Datei direkt importieren:



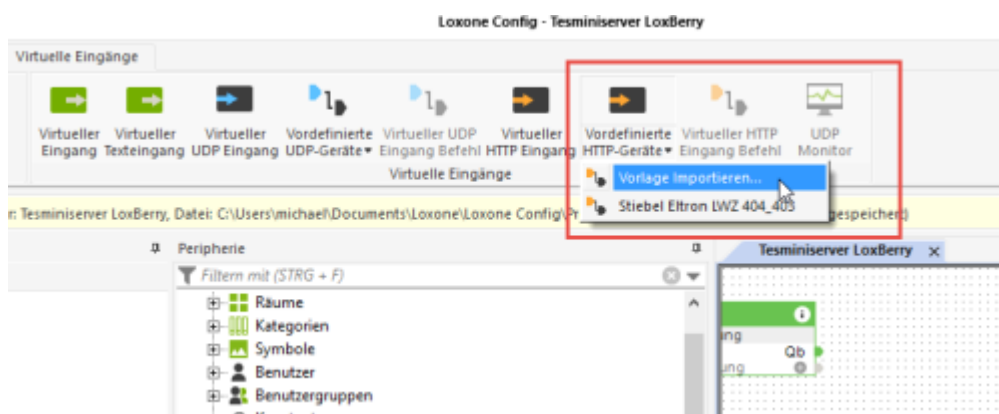
Bis LoxConfig Version 8 muss die Datei manuell in das

Verzeichnis C:\ProgramData\Loxone\Loxone Config

VERSION\Templates\VirtualIn kopiert werden. Sie muss dabei unbedingt mit **VIU\_** beginnen, damit sie erkannt wird. LoxConfig muss anschließend neu gestartet werden.

## Virtuelle HTTP Eingänge

Wenn Du die Status-Daten per HTTP-Eingang im Miniserver einlesen möchtest, musst Du dieses Template in Dein Projekt integrieren. Ab LoxConfig Version 9 kannst Du die Datei direkt importieren:



Bis LoxConfig Version 8 muss die Datei manuell in das

Verzeichnis C:\ProgramData\Loxone\Loxone Config

VERSION\Templates\VirtualIn kopiert werden. Sie muss dabei unbedingt mit **VI\_** beginnen, damit sie erkannt wird. LoxConfig muss anschließend neu gestartet werden.

# Codes

## Status Codes

| Status    | Description    | Beschreibung                       |
|-----------|----------------|------------------------------------|
| STATE 0   | Unknown        | unbekannt                          |
| STATE 1   | Initiating     | Initialisierung                    |
| STATE 2   | Sleeping       | Ladegerät von Netz getrennt        |
| STATE 3   | Idle           | Ruheposition                       |
| STATE 4   | Remote Control | Fernsteuerung aktiv                |
| STATE 5   | Cleaning       | Reinigung aktiv                    |
| STATE 6   | Returning Dock | Robot fährt zurück zur Ladestation |
| STATE 7   | Manual Mode    | Manueller Modus                    |
| STATE 8   | Charging       | Aufladen                           |
| STATE 9   | Charging Error | Ladefehler                         |
| STATE 10  | Paused         | Pausiert                           |
| STATE 11  | Spot Cleaning  | Spot Reinigung                     |
| STATE 12  | In Error       | Fehler                             |
| STATE 13  | Shutting Down  | Roboter fährt herunter             |
| STATE 14  | Updating       | Aktualisieren (Firmware)           |
| STATE 15  | Docking        | Batterie voll geladen              |
| STATE 16  | Go To          | Gehe zu Ziel                       |
| STATE 17  | Zone Clean     | Zonenreinigung aktiv               |
| STATE 18  | Room Clean     | Raumreinigung aktiv                |
| STATE 100 | Fully Charged  | Roboter voll geladen               |

## Error Codes

| Error    | Description             | Beschreibung                             |
|----------|-------------------------|------------------------------------------|
| ERROR 0  | No error                | Kein Fehler                              |
| ERROR 1  | Laser sensor fault      | Laserentfernungssensor                   |
| ERROR 2  | Collision sensor fault  | Kollisionssensor                         |
| ERROR 3  | Wheel floating          | Räder haben keinen Bodenkontakt          |
| ERROR 4  | Cliff sensor fault      | Bodensensoren reinigen                   |
| ERROR 5  | Main brush blocked      | Hauptbürste reinigen                     |
| ERROR 6  | Side brush blocked      | Seitenbürste reinigen                    |
| ERROR 7  | Wheel blocked           | Räder blockiert                          |
| ERROR 8  | Device stuck            | Robot blockiert                          |
| ERROR 9  | Dust bin missing        | Staubbehälter nicht eingesetzt           |
| ERROR 10 | Filter blocked          | Filter reinigen                          |
| ERROR 11 | Magnetic field detected | Robot an magnetischer Barriere blockiert |
| ERROR 12 | Low battery             | Batterieladung niedrig                   |
| ERROR 13 | Charging problem        | Ladefehler                               |
| ERROR 14 | Battery failure         | Batteriefehler                           |

| Error     | Description                           | Beschreibung                               |
|-----------|---------------------------------------|--------------------------------------------|
| ERROR 15  | Wall sensor fault                     | Wandsensoren reinigen                      |
| ERROR 16  | Uneven surface                        | Robot auf flachen Untergrund setzen        |
| ERROR 17  | Side brush failure                    | Problem mit Seitenbürste. Robot neu booten |
| ERROR 18  | Suction fan failure                   | Problem mit Lüfter                         |
| ERROR 19  | Unpowered charging station            | Ladestation nicht angesteckt               |
| ERROR 20  | Unknown Error                         | Robot nicht erreichbar                     |
| ERROR 21  | Laser pressure sensor problem         |                                            |
| ERROR 22  | Charge sensor problem                 |                                            |
| ERROR 23  | Dock problem                          |                                            |
| ERROR 24  | No-go zone or invisible wall detected |                                            |
| ERROR 254 | Bin full                              |                                            |
| ERROR 255 | Internal error                        |                                            |
| ERROR -1  | Unknown Error                         |                                            |

## Loxone Vorlage für Visualisierung

Vorlage (Projekt) der Status-/Error Codes (DE/EN) + simple Steuerung.

[MiRobot.Loxone](#)

## Koordinaten für Zonenreinigung ermitteln

mit dem Befehl kann man eine definierte Fläche reinigen.

[/plugins/mirobot2lox-ng/sendcmd.cgi?command=zoned\\_clean&option=\[\[30200,26280,32500,32400,1\]\]&robot=1&debug=0](#)

Die Fläche bestimmt man am einfachsten mit der Android App <https://xiaomi.flole.de/>

- links den Menüpunkt "Karte" auswählen, die Karte wird runtergeladen und angezeigt
- Schaltfläche "Hinzufügen" und dann gewünschten Bereich markieren
- "Reinigen" länger drücken, die notwendigen Koordinaten werden in die Zwischenablage kopiert
- nun noch den Befehl mit den eigenen Koordinaten ersetzen



### Hinweis

Von den neuen Roborocks scheint die Karte n FloleVac nicht angezeigt zu werden. Da bleibt aktuell nur die manuelle Variante (siehe unten). Stand: 29.082020



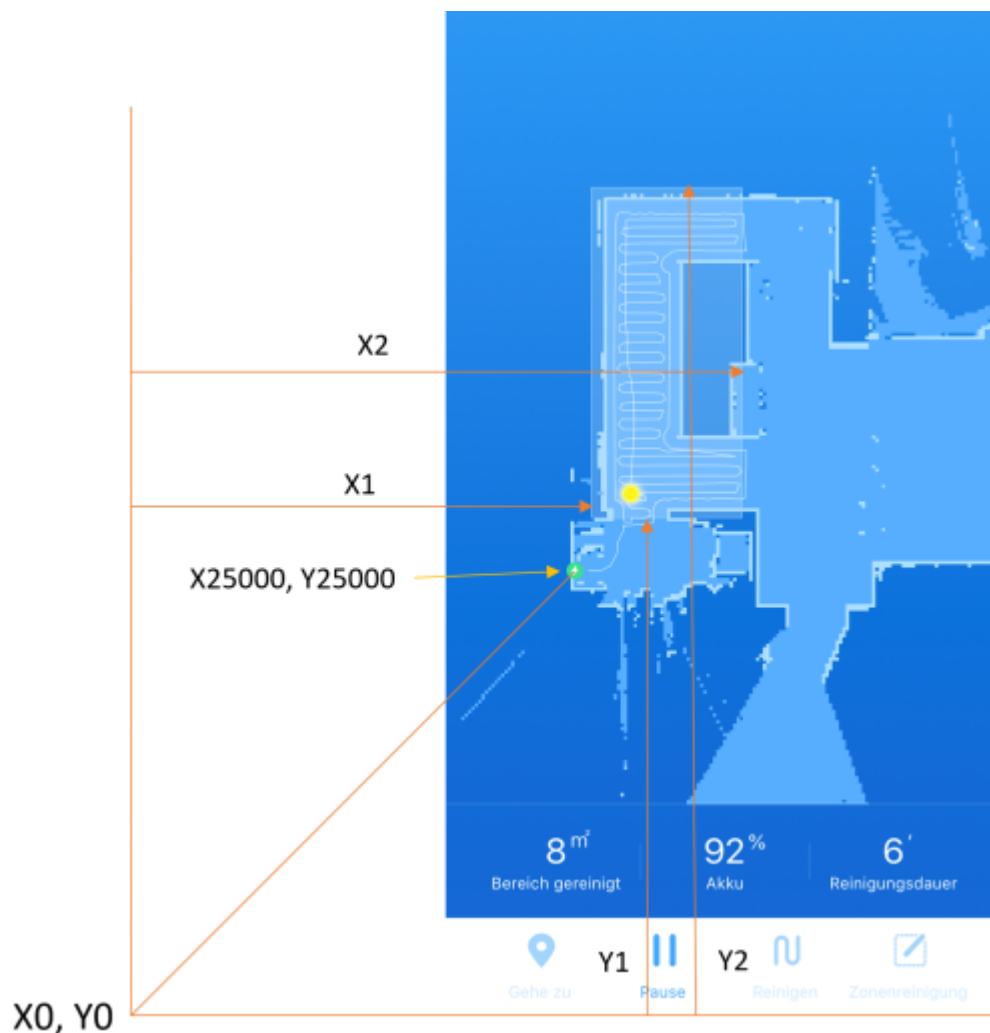
# Manuelle Koordinatenermittlung

Die Ladestation ist prinzipiell der Mittelpunkt des Roborock Universums, allerdings nicht die beste Position um die Umgebung zu scannen. Daher scheint der Roboter je nach Umfeld einen kleinen Abstand zur Ladestation zu setzen, an dem der Scan durchgeführt wird. Dies ist dann der eigentliche Mittelpunkt (25000/25000) der Karte. Das lässt sich leicht ermitteln, indem man genau diesen Punkt anfahren lässt. Der Abstand von der Ladestation zum Roboter entspricht dann dem Offset der bei allen Koordinaten auf der Etage mit eingerechnet werden sollte (wenn man das Maßband verwenden möchte).

Einzelne Punkte können mittels folgendem Befehl (am einfachsten im Browser) angefahren werden (X=28000 Y=28000 bzw. dX+3m, dY+3m)

[http://IP\\_LOXBERRY/plugins/mirobot2lox-ng/sendcmd.cgi?command=goto&option=28000%2028000&robot=1&debug=0](http://IP_LOXBERRY/plugins/mirobot2lox-ng/sendcmd.cgi?command=goto&option=28000%2028000&robot=1&debug=0)

Die "%20" in obigem Befehl zwischen den beiden Koordinaten steht für ein Leerzeichen (URLencoded)



Der Befehl für die Zonenreinigung welcher oberhalb angeführt ist ist mit der Reihenfolge **X1,Y1,X2,Y2,1** zu verwenden. Dabei ist der Startpunkt immer links unten (X1, Y1) und der Endpunkt rechts oben (X2, Y2)

Zum Testen kann wieder im Browser folgender String eingegeben werden:

[http://IP\\_LOXBERRY/plugins/mirobot2lox-ng/sendcmd.cgi?command=zoned\\_clean&option=\[\[25500,26500,27900,31300,1\]\]&robot=1&debug=0](http://IP_LOXBERRY/plugins/mirobot2lox-ng/sendcmd.cgi?command=zoned_clean&option=[[25500,26500,27900,31300,1]]&robot=1&debug=0)

Nach dem Abschluss der Zonenreinigung fährt der Roboter wieder in die Ladestation zurück und die Zone wird grafisch in der "Xiaomi Home"-App dargestellt.

## Teilautomatisierte Koordinatenermittlung

Die manuelle Koordinatenermittlung und Zonenerstellung ist sehr hilfreich, aber auch etwas nervig. Die Exceldatei ermöglicht das einfache Eintragen des Abstands vom Kartenmittelpunkt zur Ladestation und mit den gewünschten Abständen von der Ladestation zum Zielort werden die Befehle, bzw. Testlinks automatisch generiert. Diese müssen nur noch kopiert, bzw. angeklickt werden. Die IP Eurer Loxberry sollte noch vorab eintragen werden damit die Testlinks auch funktionieren. Damit ist zumindest eine Teilautomatisierung der Koordinatenermittlung möglich. Ein paar Tipps, bzw. Erfahrungen wurden noch ergänzt, in der Hoffnung dass das bei Euch dann noch besser klappt.

[1254688054.xlsx](#)

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

[https://wiki.loxberry.de/plugins/mirobot2lox\\_ng/start?rev=1700077354](https://wiki.loxberry.de/plugins/mirobot2lox_ng/start?rev=1700077354)

Last update: **2023/11/15 20:42**