



- von der aktuellen Map ein Screenshot erstellt und diesen als Hintergrundbild in Visio verwendet.
- den Zeichnungsmaßstab in Visio so eingestellt, dass die Wohnung darauf Platz findet.
- "Lineal und Gitter" (Menü Extras) konfiguriert, damit die Wohnung auf das Blatt passt. Die Ladestation befindet sich bei allen Anwendern bei ca. 25500/25500, d.h. bei 25.5m/25.5m. Die Koordinaten sind also Millimeter-Angaben.
- einen Tisch Shape auf das Blatt gezogen mit dem ich die Zone nachbilden konnte.
- damit ich die Koordinaten des Shapes erhielt, unter Menu Ansicht > Grösse- und Positionsfenster eingeschaltet. X und Y ist die linke untere Ecke des Tisches. Die obere rechte Ecke bildet sich aus X+Breite bzw. Y+Höhe. In Millimeter umgerechnet, sind dies die Koordinaten für den Befehl `app_zoned_clean`.
- mit dem mirobo Befehl `app_zoned_clean` ging es an die Feinabstimmung für das Hintergrundbild (Grösse und Position). Das Ziel ist, die gleiche Zone in der App wie auf dem Screenshot zu erstellen. Sendet man den mirobo Befehl, wird nach ein paar Sekunden die Zone auf der App dargestellt und der Robi fährt los. Den Robi kann man mit dem Befehl `pause` wieder stoppen. Er muss ja diese Zone nicht reinigen. Nach ein paar Versuchen, habe ich die richtigen Koordinaten gefunden, mit denen auf der App die gleiche Zone wie auf dem Screenshot dargestellt wird.
- Screenshot Grösse angepasst und an den richtigen Ort gezogen, sodass der Tisch über der Zone auf dem Screenshot zu liegen kommt.
- alle weiteren Zonen pro Raum erstellt und die Koordinaten herausgeschrieben.

Hier nun die Details zu den mirobo Befehlen:

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command app_goto_target [17130,26650]
```

Der Befehl `app_goto_target` benötigt zwei Integer für x und y, d.h. die Koordinaten aus dem Viso in Millimetern.

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command app_zoned_clean [[19540,24230,22900,26440,1],[17130,26440,22900,28000,1],[17130,24230,18100,26440,1],[17130,23250,23100,24230,1]]
```

Der Befehl `app_zoned_clean` hat als Argument eine Liste von maximal 5 Zonen, welche mit Komma getrennt werden. Jede Zone wird mit 5 Werten angegeben. Die ersten zwei Werte sind die Koordinaten der linken unteren Ecke (X1/Y1) und die zwei nachfolgenden Werte die Koordinaten der oberen rechten Ecke (X2/Y2). Der letzte Wert gibt an, wie oft die Zone gereinigt werden soll.

[<X1>,<Y1>,<X2>,<Y2>,<Anzahl>]

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 pause
```

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 home
```

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command find_me
```

Ein Übersicht aller Befehle gibt es hier:

<https://github.com/marcelrv/XiaomiRobotVacuumProtocol>

Über das CLI kann ich nun alle Räume einzeln reinigen lassen. Hier ein paar Beispiele:

### Wohnzimmer

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command app_zoned_clean  
[[19540,24230,22900,26440,1],[17130,26440,22900,28000,1],[17130,24230,18100,26440,1],[17130,23250,23100,24230,1]]
```

### Gang

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command app_zoned_clean [[24280,22960,32800,24180,1]]
```

### Büro

Code:

```
mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command app_zoned_clean  
[[26000,19500,32000,23020,1],[26000,19000,27040,19500,1],[31000,19000,32000,20040,1]]
```

Leider bietet das Plugin **MiRobot2Lox** diese Befehle noch nicht. Ich als Laie habe auch nicht herausfinden können, wie die udp Befehle zu erweitern sind.

Damit ich die Koordinaten als Parametern übergeben konnte, habe ich mir das Any Plugin installiert und ein paar Scripte erstellt. Ich verwalte die Koordinaten nicht in der Loxone Config sondern im Script.

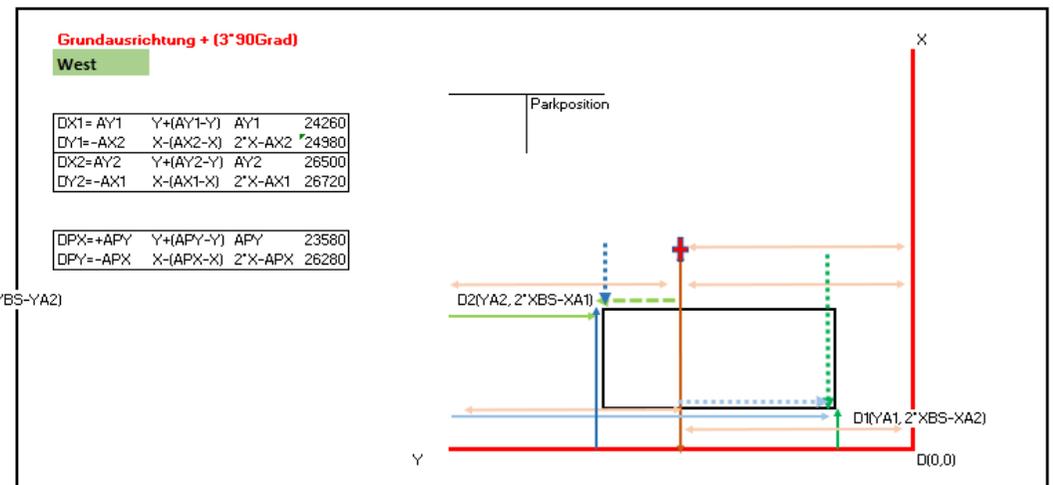
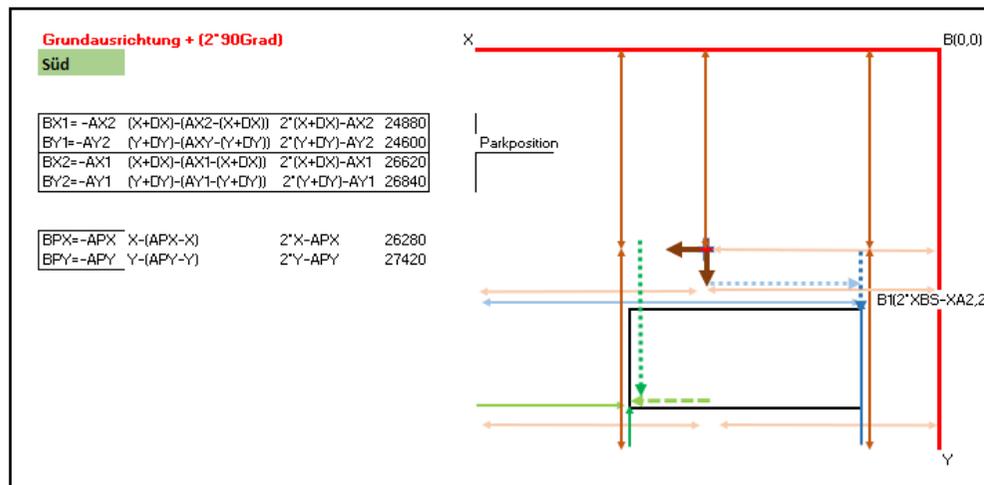
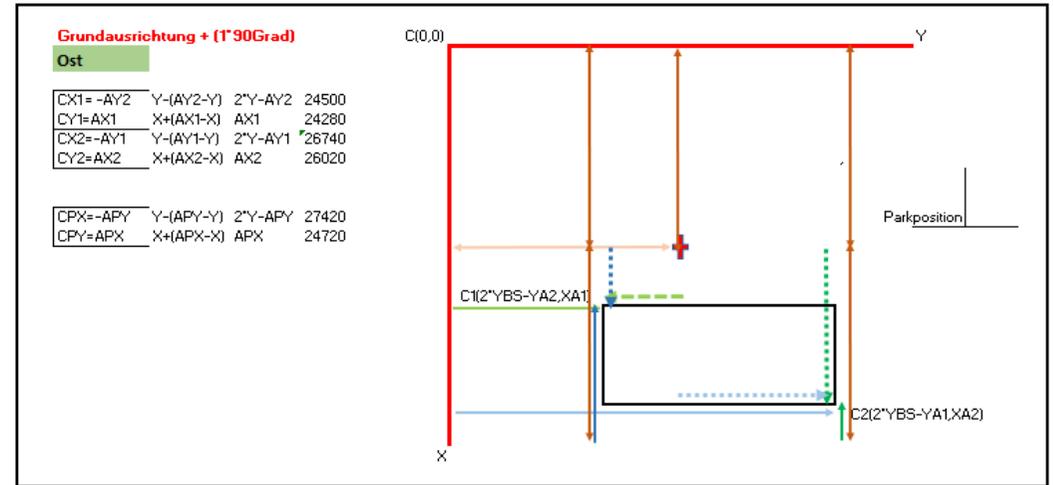
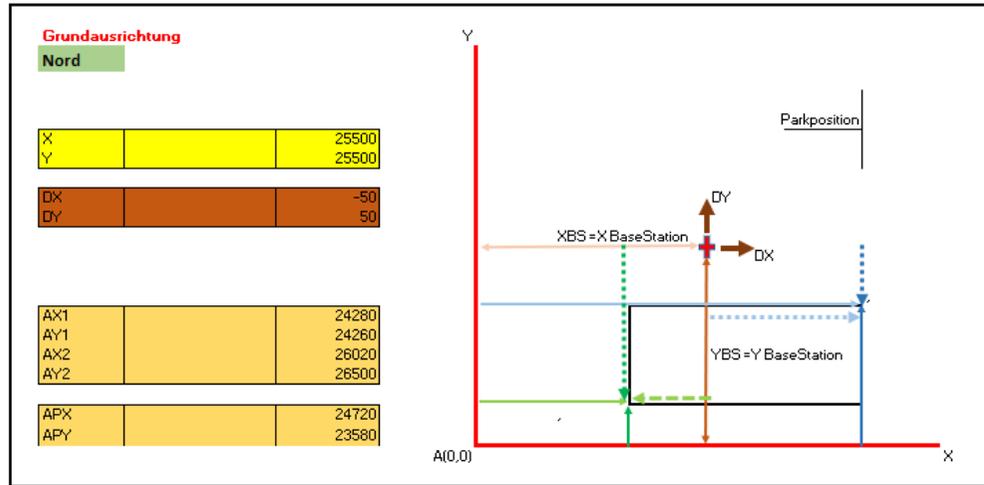
Noch ein paar Infos zu der Map und den Informationen, welche man in diversen Foren findet. Es wird immer wieder berichtet, dass die Map auf dem Handy unterschiedlich ausgerichtet wird. Dies ist gemäss meinen Erfahrungen nur sehr selten der Fall, wenn man:

- die Ladestation des Robi an einem Ort aufstellt, an dem er sich gut an Wänden beim Starten des Reinigungsvorganges orientieren kann. Scheinbar geht dies am besten, wenn er in einer Ecke steht.

Aber auch bei mir hat sich in den letzten 3 Monate die Karte einmal um 180 Grad gedreht und nie wieder zurück. Damit man die Koordinaten nicht neu erstellen muss, kann man die Orientierung der Karte als Parameter übergeben. D.h.die Scripte rechnen je nach Drehung die neuen Positionen/Zonen um.

Wie die Parameter aus der Loxone Config über das Any Plugin an die Scripte übergeben werden können, steht in den Scripten.

# Excel – Umrechnung der Koordinaten



## Any Plugin for LoxBerry V0.2.0

TCP-IN = 9095, UDP-OUT = 9096

mit putty im Loxberry einloggen

> user/pwd

ins Verzeichniss mit den Scripten wechseln

> cd /opt/loxberry/data/plugins/anyplugin/commands

In Loxone Config:

Eigenschaften (Virtueller Ausgang)	
Eigenschaft	Wert
Allgemein	
Bezeichnung	Robo_tcp
Beschreibung	
Anschluss	VQ11
Adresse	tcp://192.168.1.131:9095
<input checked="" type="checkbox"/> Verbindung nach Se...	
Trennzeichen	;
Befehl bei Verbindungs...	

Virtueller Ausgang anlegen:

Virtueller Ausgangsbefehl anlegen:

Bezeichnung: Robo Zoned\_Cleanup

Befehl bei EIN: mirobo\_zoned\_cleanup off command ./mirobo\_zoned\_cleanup <v>

Als Digitalausgang verwenden: deaktiviert

Robo New\_Map      mirobo\_new\_map off command ./mirobo\_new\_map

Robo Goto:        mirobo\_goto off command ./mirobo\_goto <v>

## Scripte in dieses Verzeichnis

mirobo\_home

```
loxberry@loxberry: ~/data/plugins/anyplugin/commands
GNU nano 2.7.4 File: mirobo_home
#!/bin/bash
export LC_ALL=C.UTF-8
export LANG=C.UTF-8
echo `mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 home`
```

mirobo\_find\_me

```
loxberry@loxberry: ~/data/plugins/anyplugin/commands
GNU nano 2.7.4 File: mirobo find me
#!/bin/bash
export LC_ALL=C.UTF-8
export LANG=C.UTF-8
echo `mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 raw_command find_me`
```

mirobo\_new\_map

```
loxberry@loxberry: ~/data/plugins/anyplugin/commands
GNU nano 2.7.4 File: mirobo new map
#!/bin/bash
export LC_ALL=C.UTF-8
export LANG=C.UTF-8
echo `mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 start`
sleep 25
echo `mirobo -d --ip 192.168.1.133 --token 3372304a685745785975573742424637 pause`
```

mirobo\_goto

```
loxberry@loxberry: ~/data/plugins/anyplugin/commands
GNU nano 2.7.4 File: mirobo_goto

#!/bin/bash
export LC_ALL=C.UTF-8
export LANG=C.UTF-8

debug=1
debug_position=1
command_send=1

#-----
# Variante 1 <b><a> als ein Parameter > b = Positions-Nr, a = Karten-Ausrichtung 1-3
# Variante 2 <b> <a> 2 einzelne Parameter
# 1. Parameter = Zonen_Nr
# 2. Parameter = Karten-Ausrichtung
#-----
# Für jede Zone müssen die folgenden Arrays angelegt werden:
# - Positions Konfiguration P_<Position#>=<Name> <AnzahlPositionen#> <Home#>
# Array-Name für eine Zonen-Konfiguration fängt mit "P_" an, gefolgt vom "Namen", welche in der Zonen-Konfiguration definiert wurde und enden mit "_" und einer fortlaufenden
# <Name>: mit diesem Namen werden die Arrays und Positionen definiert. Er muss eindeutig sein.
# <AnzahlPositionen#> Anzahl der Positionen, welche der Robi anfahren soll, damit er die zu reinigende Zone findet.
# <Home#>: Ob am Ende der Robi mit Home in Ladestation soll: 1=ja, 0=nein
# - Positions-Array P_<Name>_l=(x y s)
# pro Position wird ein Array benötigt
# Array-Name für Positionen fängt mit "P_" an, gefolgt vom "Namen", welche in der Zonen-Konfiguration definiert wurde und enden mit "_" und einer fortlaufenden "Nummer"
# x y: sind die Koordinaten der Position
# s: die Zeit in Sekunden(Sleeptime), welche gewartet wird, bis der Robi den Punkt angefahren hat und der nächste Befehl ausgeführt werden kann.
#-----
X=25500 # Ladestationposition des Robi
Y=25500
Delta_X=100 # Verschiebung Ladestationposition mit Auswirkung auf alle Zonen und Punkte
Delta_Y=50 # man kann auch direkt X oder Y ändern
#-----
P_8=(Entleeren 1 0) # Entleeren
P_Entleeren_l=(24520 22580 1)
#-----
P_6=(Complete_Map 4 1)
P_Complete_Map_1=(24720 23580 27)
P_Complete_Map_2=(35280 23580 62)
P_Complete_Map_3=(21300 23580 67)
P_Complete_Map_4=(24720 23580 32)
#-----

Position_new=""
change_position() {
    Position_AFX=$1
    Position_AFY=$2
    Ausrichtung P=$3
```

# mirobo\_zoned\_cleanup

```

loxberry@loxberry: ~/data/plugins/anyplugin/commands
GNU nano 2.7.4 File: mirobo zoned cleanup

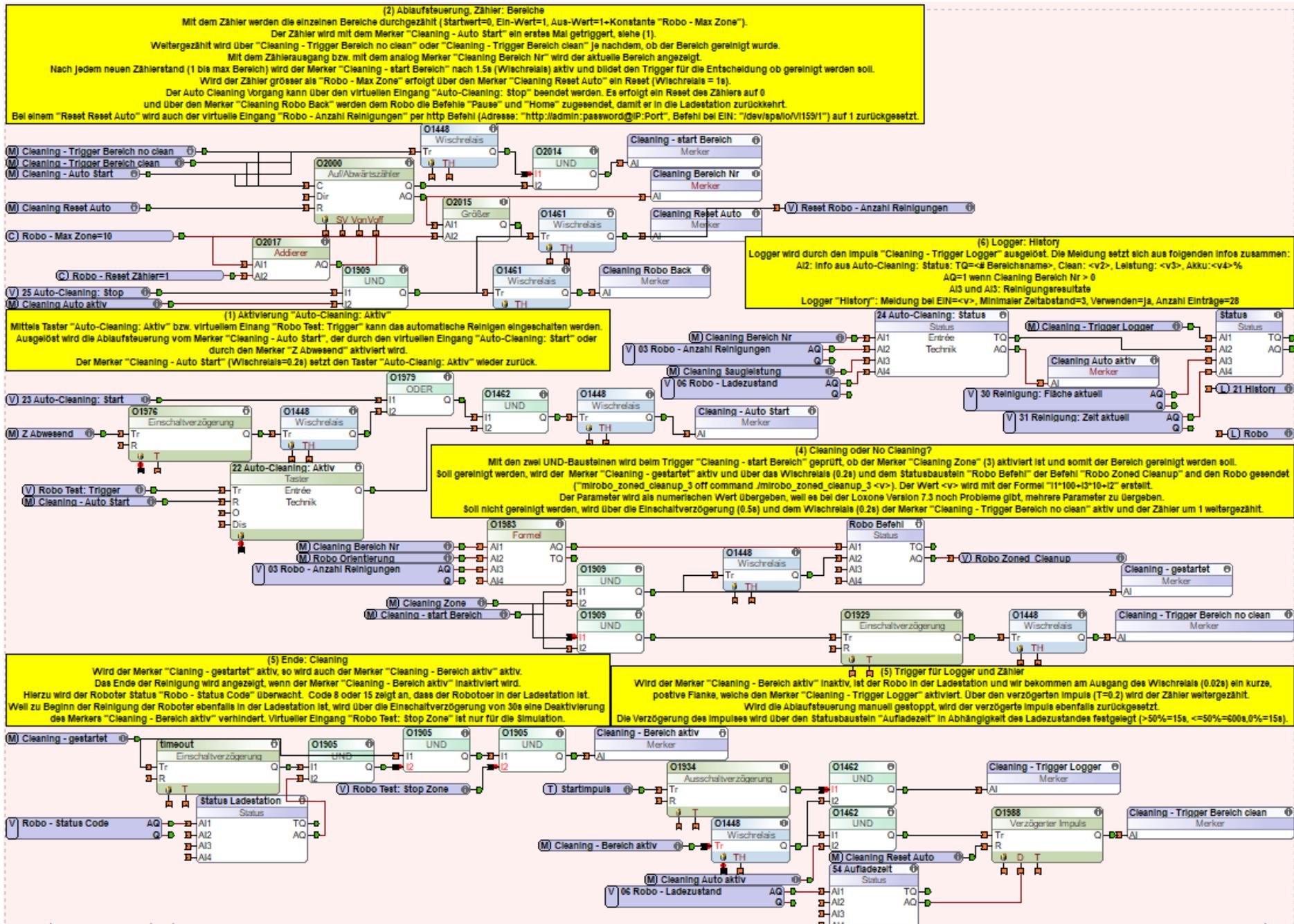
#!/bin/bash
export LC_ALL=C.UTF-8
export LANG=C.UTF-8

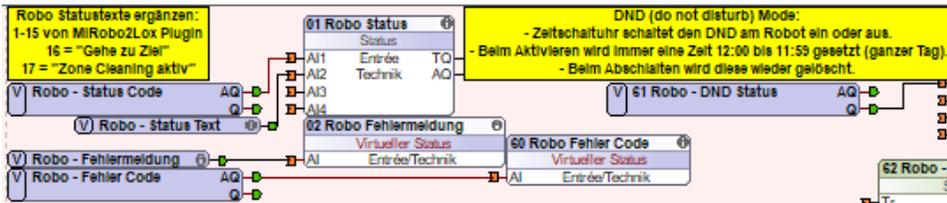
debug=0
debug_zone=1
debug_position=1
command_send=1

#-----
# Script Parameter: <c><b><a> ein Nummer mit 3-4 Stellen
#
# <a> = <1-3> Karten-Ausrichtung und entspricht der Drehung im Uhrzeigersinn von 1=0Grad, 2=90Grad, 3=180Grad, 4=270Grad
# <b> = <1-3> Anzahl Reinigungsdurchläufe
# <c> = die vorderen 1-2 Ziffern ist die zu reinigende Zonen-Nr
# oder <c> <b> <a> 3 einzelne Parameter
# 1. Parameter <c> = Zonen_Nr
# 2. Parameter <b> = Anzahl Reinigungsdurchgänge
# 3. Parameter <a> = Karten-Ausrichtung
#-----
# Für jede Zone müssen die folgenden Arrays angelegt werden:
# - Zonen Konfiguration Z_<Zone#>=<Name> <AnzahlPositionen#> <AnzahlZonen#>
# Array-Name für eine Zonen-Konfiguration fängt mit "Z_" an, gefolgt vom "Namen", welche in der Zonen-Konfiguration definiert wurde und enden mit "_" und einer fortlaufenden "Nummer"
# <Name>: mit diesem Namen werden die Arrays und Positionen definiert. Er muss eindeutig sein.
# <AnzahlPositionen#> Anzahl der Positionen, welche der Robi anfahren soll, damit er die zu reinigende Zone findet.
# <AnzahlZonen#>: Anzahl Zonen, welche der Robi reinigen soll (max. 4)
# - Positions-Array P_<Name>_1=(x y s)
# pro Position wird ein Array benötigt
# Array-Name für Positionen fängt mit "P_" an, gefolgt vom "Namen", welche in der Zonen-Konfiguration definiert wurde und enden mit "_" und einer fortlaufenden "Nummer"
# x y: sind die Koordinaten der Position
# s: die Zeit in Sekunden(Sleeptime), welche gewartet wird, bis der Robi den Punkt angefahren hat und der nächste Befehl ausgeführt werden kann.
# - Zonen-Array Z_<Name>_1=(x1 y1 x2 y2)
# pro Zone wird ein Array benötigt
# Array-Name für Zonen fängt mit "Z_" an, gefolgt vom "Namen", welche in der Zonen-Konfiguration definiert wurde und enden mit "_" und einer fortlaufenden "Nummer"
# x1, y1, x2, y2: sind die Koordinaten der Zone
#-----
X=25500 # Ladestationposition des Robi
Y=25500
Delta_X=100 # Verschiebung Ladestationposition mit Auswirkung auf alle Zonen und Punkte
Delta_Y=50 # man kann auch direkt X oder Y ändern
#-----
Z_1=(EN_GA_RE 0 3)
Z_EN_GA_RE_1=(24280 24260 26020 26500) # Entrée: Koordinaten für eine Zone
Z_EN_GA_RE_2=(24280 22960 32800 24180) # Gang:
Z_EN_GA_RE_3=(28520 24260 29520 25190) # Reduit:
#-----
Z_2=(KE 0 1)
Z_KE_1=(19900 19000 25200 23600) # Küche/Essen:
#-----
Z_3=(WZ 2 4) # Wohnzimmer

```

# Loxone Config (Version 7.1.9.30)

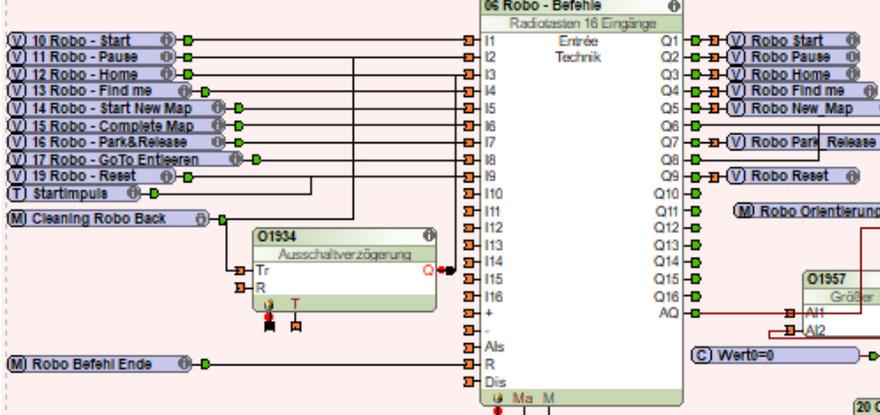




**Saugleistung-Werte vereinfachen:**  
 Das Plugin MiRobo2Lox sendet bei Änderung über das Plugin oder über die Xiaomi App nicht die gleichen Werte:  
 - mittels Xiaomi App: 101, 102, 103, 104 -> 1, 2, 3, 4  
 - mittels Plugin: 38, 60, 77, 90 -> 1, 2, 3, 4  
 Änderungen werden per UDP Befehl an das MiRobo2Lox Plugin gesendet  
 -> R1\_Mi\_FAN\_Q, R1\_Mi\_FAN\_B, R1\_Mi\_FAN\_T, R1\_Mi\_FAN\_F

**Einzelne Robo Befehle zusammenfassen**  
 Wird ein Ausgang der Radiotasten aktiv, wird ein Befehl an den Robo gesendet. Nach 1s wird der Baustein wieder zurückgesetzt.  
 Bei Dekativierung des Auto Clean Vorganges wird über den Merker "Cleaning Robo Back" der Robo gestoppt (Pause) und 3s später in die Ladestation zurückgeschickt (Home).

**Befehle:**  
 Robo - Reset: Startimpuls reinitialisiert das MiRobo2Lox Plugin.  
 Robo - start New Map: startet eine Reinigung für 20s, damit die Karte neu aufgebaut und dessen Orientierung festgelegt wird. Eine neu Orientierung kann per Radiotasten "Robo - Map Orientierung" angepasst werden.  
 Robo - Complete Map: Es werden Positionen per GoTo angefahren, damit eine vollständige Karte erstellt wird (nicht mehr notwendig)  
 Robo - GoTo Entleeren: Damit wird ein Zielpunkt per GoTo Befehl angefahren



Technik

- 01 Robo Status Akku voll geladen
- 02 Robo Fehlermeldung Keine Fehler
- 03 Robo - Anzahl Reinigungen 1
- 04 Robo - Saugleistung > Max
- 05 Robo - Map Orientierung > Süd
- 06 Robo - Befehle > Inaktiv
- Manuell >
- 21 History > 19.09.2018 07:46:35

Technik

- 22 Auto-Cleaning: Aktiv
- 23 Auto-Cleaning: Start
- 24 Auto-Cleaning: Status Inaktiv
- 24 Auto-Cleaning: Stop
- 30 Reinigung: Fläche aktuell 22m2
- 31 Reinigung: Zeit aktuell 17'
- 40 Care Hauptbürste 89%
- 41 Care Seitenbürste 84%

Technik

- 50 Cleaning History Count 298
- 51 Cleaning History Area 1909m2
- 52 Cleaning History Time 1923'
- 53 Cleaning History last Cleaning Time 5165'
- 60 Robo Fehler Code 0
- 61 Robo - DND Status 1
- 62 Robo - DND Timer > Aktiv
- Status Cleaning Resultat ^
  - Fläche: 22m2, Zeit: 17', Anzahl: 1, Sauger: 4

04 Robo - Saugleistung

- Quiet
- Balanced
- Turbo
- Max

05 Robo - Map Orientierung

- Nord
- Ost
- Süd
- West

06 Robo - Befehle

- Start
- Pause
- Home
- Find me
- Start New Map
- Complete Map
- Park&Release
- GoTo Entleeren
- Inaktiv

20 Cleaning Bereiche

Szenen Manuell

- 1 EN & GA & RE
- 2 K/E
- 3 WZ
- 4 SZ
- 5 BZ & GB
- 6 BO/BW
- 7 GZ
- 8 Ofen
- 9 Sofa

20 History

14. SEPTEMBER 2018

- 10:32:17 Status: 6 BO/BW, Auto:1, Fläche: 19m2, Zeit: 20', Power: 4, Akku:35%
- 09:59:52 Status: 5 BZ & GB, Auto:1, Fläche: 13m2, Zeit: 12', Power: 4, Akku:46%
- 09:35:04 Status: 4 SZ, Auto:1, Fläche: 18m2, Zeit: 15', Power: 4, Akku:47%
- 09:16:58 Status: 3 WZ, Auto:1, Fläche: 22m2, Zeit: 23', Power: 4, Akku:57%
- 08:50:56 Status: 2 K/E, Auto:1, Fläche: 22m2, Zeit: 22', Power: 4, Akku:74%
- 08:27:17 Status: 1 EN & GA & RE, Auto:1, Fläche: 16m2, Zeit: 15', Power: 4, Akku:89%
- 08:10:39 Cleaning - Auto Start:
- 07:48:41 Status: 10 Küche, Auto:1, Fläche: 8m2, Zeit: 6', Power: 4, Akku:96%
- 07:40:48 Cleaning - Auto Start:
- 07:33:42 Status: 10 Küche, Auto:1, Fläche: 0m2, Zeit: 0', Power: 4, Akku:100%
- 07:32:35 Cleaning - Auto Start:
- 07:31:55 Robo Reset:
- 07:31:54 Robo Reset:

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI1	==	1	-	==	0	-	==	0	-	==	0	In Arbeit	1 EN & GA & RE, Clean: <v2>, Leistung: <v3>, Akk: <v4>%	1
AI1	==	2	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	3	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	4	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	5	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	6	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	7	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	8	-	==	0	-	==	0	-	==	0	In Arbeit		1
AI1	==	9	-	==	0	-	==	0	-	==	0	In Arbeit		9 Sofa, Clean: <v2... 1
AI1	==	10	-	==	0	-	==	0	-	==	0	In Arbeit		10 Küche, Clean: ... 1
AI1	==	11	-	==	0	-	==	0	-	==	0	In Arbeit		11 tbd, Clean: <v2... 1
AI1	==	12	-	==	0	-	==	0	-	==	0	In Arbeit		12 tbd, Clean: <v2... 1
-	==	0	-	==	0	-	==	0	-	==	0	Inaktiv	inaktiv	0

Status Auto-Cleaning Status

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI1	==	1	-	==	0	-	==	0	-	==	0	Nicht zu...	<v2>, Fläche: <v3>m2, Zeit: <v4>'	
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...		

Status Status

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI2	==	1	-	==	0	-	==	0	-	==	0	Nicht zu...		<v1>
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...		

Status Robo Befehl

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI1	==	8	-	==	0	-	==	0	-	==	0	Nicht zu...	Robo in Docking St...	1
AI1	==	15	-	==	0	-	==	0	-	==	0	Nicht zu...	Robo in Docking St...	1
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...	Robo aktiv	0

Status Ladestation

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI1	>	60	-	==	0	-	==	0	-	==	0	Nicht zu...	15s	15
AI1	<=	50	AI1	>	0	-	==	0	-	==	0	Nicht zu...	600s	600
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...		15

Status Aufladzeit

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statustext	Statuswert
AI1	==	38	-	==	0	-	==	0	-	==	0	Nicht zu...		1
AI1	==	60	-	==	0	-	==	0	-	==	0	Nicht zu...		2
AI1	==	77	-	==	0	-	==	0	-	==	0	Nicht zu...		3
AI1	==	90	-	==	0	-	==	0	-	==	0	Nicht zu...		4
AI1	==	101	-	==	0	-	==	0	-	==	0	Nicht zu...		1
AI1	==	102	-	==	0	-	==	0	-	==	0	Nicht zu...		2
AI1	==	103	-	==	0	-	==	0	-	==	0	Nicht zu...		3
AI1	==	104	-	==	0	-	==	0	-	==	0	Nicht zu...		4
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...		0

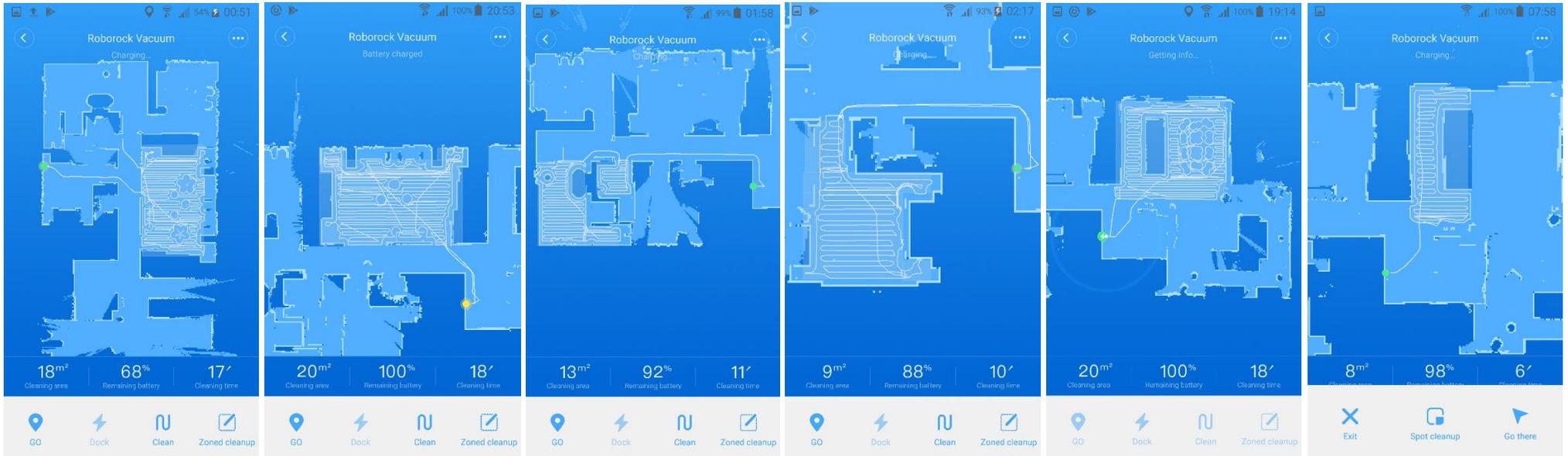
Status Saugleistung

## Status Robo GoTo

Status bearbeiten

I	V1	Wert	I	V2	Wert	I	V3	Wert	I	V4	Wert	Symbol	Statuswert	Statuswert
AI1	==	1	-	==	0	-	==	0	-	==	0	Nicht zu...		<v2>
-	==	0	-	==	0	-	==	0	-	==	0	Nicht zu...		

## Map-Ausrichtung: Bereiche mit den dazugehörigen Zonen:



Ost Büro (1 Zone)

Süd Büro (3 Zonen)

Süd Bad+WC+Gästebad (3 Z)

Süd Gästezimmer (4 Z)

Nord Küche/Essen (1 Z)

Nord Küche (1 Z)

- geht nicht in die nassen

Duschen

- geht nicht unters Sofa

- nur zwischen Kombination

und Kochinsel