

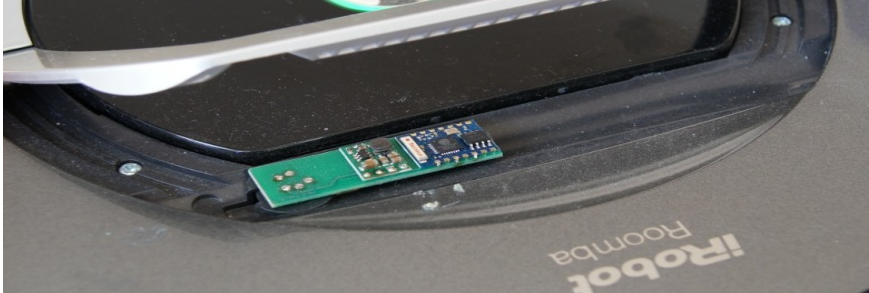
# Roomba – WiFi

howl@gmx.de

Inbetriebnahme:

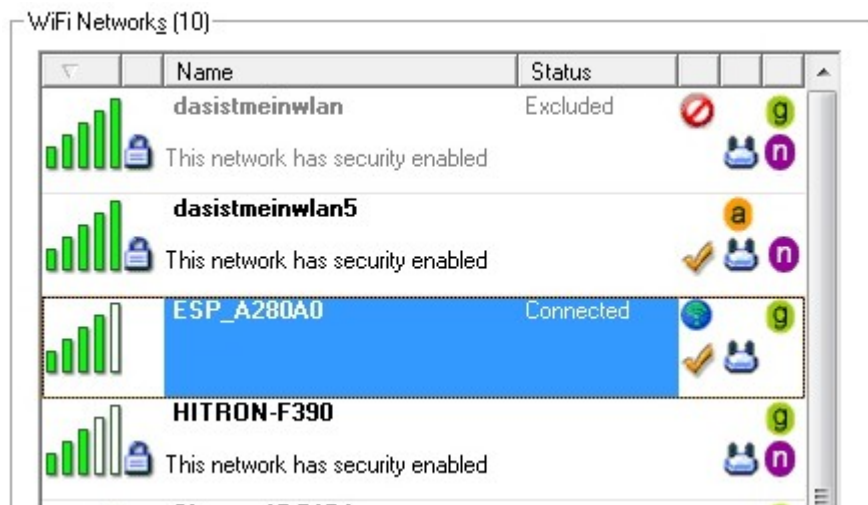
Das Modul wird wie folgt auf den Roomba aufgesteckt.

Dabei sollte der Roomba angehoben sein, da durch das aufstecken des Moduls eine Reaktion des Roombas erfolgen kann. Alternativ kann die Batterie des Roomba entfernt werden.



(1)

Wird das Modul das erste mal in Betrieb genommen geht es selbst in den Accesspoint Mode und wird als Netzwerk ESP\_XXXXXX unter den verfügbaren Netzwerken angezeigt:



Ihr Rechner muss auf DHCP eingestellt werden.

Verbinden Sie dann Ihren Rechner mit diesem Netzwerk, dafür wird kein Passwort benötigt.

Vom Modul erhält ihr Rechner die DHCP-Adresse 192.168.4.100

(2)

Öffnen Sie einen Browser und geben Sie die folgende Adresse ein:

<http://192.168.4.1>

Sie werden mit dem Modul verbunden:



The screenshot shows a web browser window with the address bar displaying 'http://192.168.4.1/'. The page title is 'Roomba WiFi'. Below the title, the email address 'HoWL@gmx.de' is displayed. The main heading is 'Bitte MasterKennwort eingeben'. A note in parentheses states '(keine Sonderzeichen, keine Leerzeichen, nur Zahlen und Buchstaben)'. There is a text input field labeled 'Masterkennwort:' and a 'senden' button.

Das Masterkennwort muß für Änderungen an den WLAN Einstellungen immer angegeben werden. Benutzen Sie ein ausreichend langes Kennwort, es sollte nicht „Roomba“ oder Ihre Netzwerk-SSID sein.

Sollten Sie es vergessen haben kann es nur durch einen Hardware Reset am Modul zurückgesetzt werden.(5)

(3)

Nach der Eingabe des MasterKennwortes werden Sie aufgefordert Ihre WLAN-Parameter und das Masterkennwort erneut einzugeben:



The screenshot shows the same web browser window, but the page has advanced to the next step. The heading is still 'Roomba WiFi'. The email address 'HoWL@gmx.de' is shown. Below it, there are three text input fields: 'WLAN SSID:', 'WLAN Passwort:', and 'MasterKennwort:'. A 'senden' button is located at the bottom right of the form.

Wurden die Daten abgesendet erscheint diese Bestätigung:

## Roomba WiFi

HoWL@gmx.de

---

WLAN Parameter gespeichert.

Kann sich der Roomba mit Ihrem Netzwerk verbinden erhält er von dort per DHCP eine neue IP-Adresse

Rufen Sie die Webseite mit dieser neuen IP auf, z.B. <http://192.168.0.100>

Das Modul versucht jetzt in Ihrem Netz eine Verbindung herzustellen.

Dazu muß Ihr Router so eingestellt sein, das er neue WLAN Geräte zulässt und DHCP Adressen vergibt.

In Ihrem Router sehen Sie dann ein neues Gerät, bei einer FritzBox könnte das z.B. so aussehen:



Sie sollten dem Eintrag einen DNS Namen vergeben, z.b. „Roomba“ und die Adressvergabe so einstellen, das immer die gleiche Adresse vergeben wird (Routerspezifisch).

Sollte sich das Modul nicht mit Ihrem Netz verbinden können bleibt es im Accesspoint Mode und Sie müssen Punkt 1 wieder ausführen. Das Masterkennwort bleibt dabei erhalten.

Es kann zu Problemen kommen, wenn Ihre SSID oder das Kennwort Sonderzeichen/Leerzeichen enthält. Sie sollte Ihre Netzwerkparameter dann ggfs. ändern.

Verbinden Sie Ihren Rechner jetzt wieder mit Ihrem Netzwerk.

(4)

Unter der neuen Wlan-IP ist das Modul jetzt erreichbar, bei Ihnen unter der vom Router vergebenen IP-Adresse:

http://192.168.0.129/ x +

192.168.0.129

## Roomba WiFi

HoWL@gmx.de

---

Version: Jun 21 2015 16:04:45  
Uptime: 0 Tag(e), 0 Stunde(n), 0 Minute(n), 41 Sekunde(n)

Konfiguration WLAN

---

Clean  
Dock  
Stop  
Wartung

Achtung: Roomba fährt ca. 130cm rückwärts !

Sensoren aktualisieren

Status	4
Spannung	15725 mV
Temperatur	38 C
Kapazitaet	2696 mAh
Ladung	2629 mAh
Strom	-188 mA

(5)

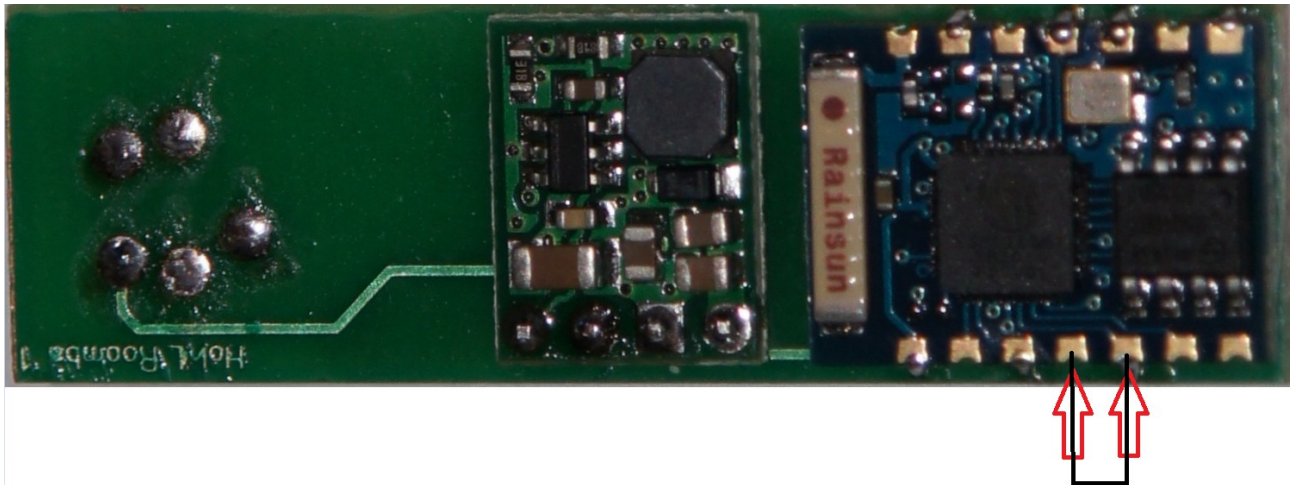
### **Master-Reset**

Durch eine Verbindung von 2 Lötunkten am Modul können Sie das Modul auf Werkseinstellungen zurücksetzen.

Verbinden Sie im eingesteckten Zustand diese beiden Lötunkte für ca. 5 Sekunden.

Danach sollte sich das Modul wieder als Accesspoint finden lassen.

Alle Einstellungen wie Masterkennwort und WLAN Parameter werden zurückgesetzt.



## **Steuerung über die HTTP-Schnittstelle**

Der Roomba lässt sich im Netzwerk fernsteuern und kann z.B. von in einer Hausautomation verwendet werden.

Die Befehle werden als URL abgesendet. Die IP-Adresse ersetzen Sie mit der vom Router zugewiesenen Adresse bzw. mit dem DNS Namen.

Als Antwort wird immer zurückgegeben „OK:ROOMBA:CMD“

Clean: <http://192.168.0.100/cmd=207>

Dock: <http://192.168.0.100/cmd=206>

Stop: <http://192.168.0.100/cmd=205>

**Achtung ! Bei den nachfolgenden Funktionen sind alle Sicherheitsfunktionen des Roomba ausgeschaltet !**

**Fährt der Roomba gegen ein Hindernis stoppt er nicht, Absturzsensoren sind ebenfalls deaktiviert !**

Der Roomba fährt die angegebene Wegstecke in cm

130cm rückwärts: <http://192.168.0.100/cmd=202:130>

300 cm vorwärts: <http://192.168.0.100/cmd=201:300>

Drehung in Grad

nach rechts 90 Grad <http://192.168.0.100/cmd=203:90>

nach linkes komplette Drehung <http://192.168.0.100/cmd=204:360>

Über die Schnittstelle können auch weitere Befehle des Roomba gesendet werden.

Das Format ist immer Befehl und danach die Parameter mit Doppelpunkt getrennt.

Übersicht z.b. hier:

[http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface\\_v2.pdf](http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf)

## **Statusdaten auslesen**

Der Roomba ermittelt alle 30 Sekunden die Statusdaten „Power“  
Diese können wie folgt abgerufen werden:

<http://192.168.0.100/cmd=142:3>

Der Roomba liefert einen String zurück:

OK:ROOMBA:3:4:15003:-203:38:2960:2967

Die einzelnen Werte sind:

3	Status der Gruppe „Power“
4	Ladezustand
15003	Spannung des Akku in Millivolt
-203	Stromverbrauch (negativ) bzw. Ladestrom (positiv)
38	Akku Temperatur in Grad Celsius
2960	Ladezustand
2967	Gesamtkapazität des Akku

Ladezustand:

0	keine Ladung
1	Konditionierung
2	Ladung
3	Erhaltungsladung
4	warten
5	Fehler beim Laden

## **Integration in openHAB**

### **items:**

// Statusdaten

Group grp\_roomba

Number roomba\_voltage (grp\_roomba)

Number roomba\_temperature (grp\_roomba)

Number roomba\_charging\_state (grp\_roomba)

Number roomba\_current (grp\_roomba)

Number roomba\_b\_charge (grp\_roomba)

Number roomba\_b\_capacity (grp\_roomba)

Number roomba\_capacity\_percent (grp\_roomba)

//Einstellungen

Group grproombaset

Switch roomba\_work\_mo (Einstellungen)

Switch roomba\_work\_di (Einstellungen)

Switch roomba\_work\_mi (Einstellungen)

Switch roomba\_work\_do (Einstellungen)

Switch roomba\_work\_fr (Einstellungen)

Switch roomba\_work\_sa (Einstellungen)

Switch roomba\_work\_so (Einstellungen)

Number roomba\_work\_hour (Einstellungen)

Number roomba\_work\_minute (Einstellungen)

### **rrd4j.persist**

Einstellungen\* : strategy = everyMinute, everyChange, restoreOnStartup

grp\_roomba\* : strategy = everyMinute, everyChange, restoreOnStartup



## sitemap:

```
Text label="Roomba" icon="roomba" {
    Frame label="Aktion" {
        Switch item=Roomba_Clean label="Clean" mappings=[ON="Start"]
        Switch item=Roomba_Dock label="Dock" mappings=[ON="Start"]
        Switch item=Roomba_Stop label="Stop" mappings=[ON="Stop"]
        Switch item=ROOMBA_OUT label="Wartung" mappings=[ON="Start"]
    }

    Frame label="Status" {
        Text item=roomba_charging_state label="Status [MAP(roomba.map):%s]"
        Text item=roomba_voltage label="Akku Spannung[%0.0f mV]"
        visibility=[roomba_charging_state!="10"] valuecolor=[>13000="green",<13000="red"]
        Text item=roomba_temperature label="Akku Temperatur[%0.0f °C]"
        icon="temperature" visibility=[roomba_charging_state!="10"]
        valuecolor=[<45="green",>=45="red"]
        Text item=roomba_current label="Strom [%0.0f mA]"
        visibility=[roomba_charging_state!="10"] valuecolor=[>0="green",<0="red"]
        Text item=roomba_b_capacity label="Kapazität [%0.0f mAh]"
        visibility=[roomba_charging_state!="10"]
        Text item=roomba_b_charge label="Ladung [%0.0f mAh]"
        visibility=[roomba_charging_state!="10"] valuecolor=[>1348="green",<1348="orange"]
        Text item=roomba_capacity_percent label="Ladezustand [%0.0f %%"
        visibility=[roomba_charging_state!="10"] valuecolor=[>50="green",<50="red"]
    }

    Frame label=" " {
        Text label=" Roomba Einstellungen" icon="settings" {
            Frame label="Startzeit" {
                Setpoint item=roomba_work_hour label="Stunde [%d]" minValue=00
                max=23 step=1 icon="clock"
                Setpoint item=roomba_work_minute label="Minute [%d]" minValue=00
                max=59 step=5 icon="clock"
            }

            Frame label="Wochentage" {
                Switch item=roomba_work_mo label="Montag"
                Switch item=roomba_work_di label="Dienstag"
                Switch item=roomba_work_mi label="Mittwoch"
                Switch item=roomba_work_do label="Donnerstag"
                Switch item=roomba_work_fr label="Freitag"
                Switch item=roomba_work_sa label="Samstag"
                Switch item=roomba_work_so label="Sonntag"
            }
        }
    }
}
```

Das sieht dann so aus:

Haupt...

Roomba

Aktion

 Clean 

Start

 Dock 

Start

 Stop 

Stop

 Wartung 

Start

Status

Status

Erhaltung (3)

Akku Spannung

16281 mV

 Akku Temperatur

39 °C

Strom

39 mA

Kapazität

2696 mAh

Ladung

2696 mAh

Ladezustand

100 %

 Roomba Einstellungen >

©2010-2014 openHAB.org

Roomba

Roomba Einstellungen

Home

Startzeit

 Stunde

14

▼ ▲

 Minute

30

▼ ▲

Wochentage

 Montag

I

 Dienstag

O

 Mittwoch

I

 Donnerstag

O

 Freitag

I

 Samstag

O

 Sonntag

O

©2010-2014 openHAB.org

## rules:

```
import java.util.Arrays
import java.util.List
import java.util.ArrayList

var Timer timer_roomba = null

var String Roomba_IP = "http://192.168.0.129/"
var String myROOMBA_CLEAN = "cmd=207"
var String myROOMBA_START = "cmd=128"
var String myROOMBA_WAKEUP = "cmd=128"
var String ROOMBA_CONTROL = "cmd=130"

var String myROOMBA_STOP = "cmd=205"
var String myROOMBA_DOCK = "cmd=206"
var String myROOMBA_OUT = "cmd=202:130" //130cm

// lambda expression
val org.eclipse.xtext.xbase.lib.Functions$Function1 httpLogic = [
    String url |

    var String tmp = ""
    var Integer timeout = 0

    while (timeout < 10)
    {
        tmp = sendHttpRequest(url)
        logInfo("rule","(timeout=" + timeout + ") " + url + " --> " + tmp)
        if (tmp.contains("OK")) return 1
        timeout = timeout + 1
        Thread.sleep(5000)
    }
]

#####
rule "Roomba Clean"
when Item Roomba_Clean changed to ON
then
    postUpdate(Roomba_Dock,OFF)
    httpLogic.apply(Roomba_IP + myROOMBA_CLEAN)

    postUpdate(Roomba_Clean,OFF)
end
```

```
#####  
rule "Roomba Dock"  
when  
    Item Roomba_Dock changed to ON  
then  
    postUpdate(Roomba_Clean,OFF)  
    ihttpLogic.apply(Roomba_IP + myROOMBA_DOCK)  
  
    postUpdate(Roomba_Dock,OFF)  
end
```

```
#####  
rule "Roomba Stop"  
when  
    Item Roomba_Stop changed to ON  
then  
    postUpdate(Roomba_Dock,OFF)  
    postUpdate(Roomba_Clean,OFF)  
    httpLogic.apply(Roomba_IP + myROOMBA_STOP)  
    postUpdate(Roomba_Stop,OFF)  
end
```

```
#####  
rule "Roomba Reinigung 130cm rückwärts fahren"  
when  
    Item ROOMBA_OUT received command ON  
then  
    sendHttpRequest(Roomba_IP + ROOMBA_CONTROL)  
    sendHttpRequest(Roomba_IP + myROOMBA_OUT)  
    postUpdate(ROOMBA_OUT,OFF)  
end
```

## Statusabfrage:

```
#####
```

```
rule "Roomba Status Abfrage"
```

```
when
```

```
    Time cron "0/30 * * * * ?"
```

```
then
```

```
    var String url
```

```
    var String tmp
```

```
    var String[] buffer
```

```
    var Number current
```

```
    var Number roomba_state
```

```
    var Integer wert
```

```
    url = Roomba_IP + "cmd=142:3"
```

```
    tmp = sendHttpRequest(url)
```

```
    logInfo("rule","Roomba Status: " + url + " --> " + tmp)
```

```
    if (tmp.contains("OK")) {
```

```
        buffer= tmp.split(":")
```

```
        roomba_state = Integer.parseInt(buffer.get(3)) // Status
```

```
        if (roomba_state > 0)
```

```
        {
```

```
            wert = Integer.parseInt(buffer.get(4))
```

```
            roomba_voltage.postUpdate(wert)
```

```
            current = Integer.parseInt(buffer.get(5))
```

```
            roomba_current.postUpdate(current)
```

```
            wert = Integer.parseInt(buffer.get(6))
```

```
            iroomba_temperature.postUpdate(wert)
```

```
            wert = Integer.parseInt(buffer.get(7))
```

```
            roomba_b_charge.postUpdate(wert)
```

```
            wert = Integer.parseInt(buffer.get(8))
```

```
            roomba_b_capacity.postUpdate(wert)
```

```
            roomba_percent = (roomba_b_charge.state as DecimalType).intValue * 100 /
```

```
(roomba_b_capacity.state as DecimalType).intValue
```

```
            if (roomba_percent > 0 && roomba_percent <=100)
```

```
postUpdate(roomba_capacity_percent,roomba_percent)
```

```
            if (current < -255) roomba_state = 6;
```

```
            roomba_charging_state.postUpdate(roomba_state)
```

```
        }
```

```
    }
```

```
end
```

Roomba zur eingestellten Zeit starten:

Kurz nach Mitternacht oder beim Systemstart wird geprüft ob an diesem Tag eine Reinigung stattfinden soll und die Minuten bis dahin ausgerechnet und ein Timer gestartet.

```
#####
```

```
rule "Roomba Startzeit bestimmen"
```

```
when Time cron "0 0 4 * * ?" or
```

```
    Item roomba_work_hour changed or
    Item roomba_work_minute changed or
    Item roomba_work_mo changed or
    Item roomba_work_di changed or
    Item roomba_work_mi changed or
    Item roomba_work_do changed or
    Item roomba_work_fr changed or
    Item roomba_work_sa changed or
    Item roomba_work_so changed or
    Item System started
```

```
then
```

```
    var int akt_minuten
```

```
    var int roomba_sollMinute
    var int roomba_sollStunde
    var int roomba_minuten
    var int warte_minuten
    var int dayofweek
```

```
// nach Neustart warten bis Variablen initialisiert
```

```
while (roomba_work_hour.state == Uninitialized || roomba_work_minute.state == Uninitialized ||
    roomba_work_mo.state == Uninitialized ||
    roomba_work_di.state == Uninitialized ||
    roomba_work_mi.state == Uninitialized ||
    roomba_work_do.state == Uninitialized ||
    roomba_work_fr.state == Uninitialized ||
    roomba_work_sa.state == Uninitialized ||
    roomba_work_so.state == Uninitialized)
{
    println("Roomba wartet nach Neustart auf Variablen")
    Thread.sleep(60000)
}
```

```
dayofweek = now.getDayOfWeek()
```

```
// Timer loeschen falls schon gesetzt
```

```
if (timer_roomba != null)
{
    timer_roomba.cancel
    timer_roomba=null
}
```

```
if ((roomba_work_mo.state == ON && dayofweek == 1) ||
    (roomba_work_di.state == ON && dayofweek == 2) ||
    (roomba_work_mi.state == ON && dayofweek == 3) ||
    (roomba_work_do.state == ON && dayofweek == 4) ||
```

```

(roomba_work_fr.state == ON && dayofweek == 5) ||
(roomba_work_sa.state == ON && dayofweek == 6) ||
(roomba_work_so.state == ON && dayofweek == 7))
{

    roomba_sollMinute = (roomba_work_minute.state as DecimalType).intValue
    roomba_sollStunde = (roomba_work_hour.state as DecimalType).intValue
    roomba_minuten = (roomba_sollStunde * 60) + roomba_sollMinute

    // aktuelle Minute
    akt_minuten = (now.getHourOfDay() * 60) + now.getMinuteOfHour()

    warte_minuten = 0
    if (roomba_minuten > akt_minuten)
    {
        warte_minuten = roomba_minuten - akt_minuten

        println("roomba start in " + warte_minuten + " Minuten")

        timer_roomba = createTimer(now.plusMinutes(warte_minuten)) []
        {
            println("roomba timer loest aus")

            if (Anw_Status.state == 1) sendCommand(Roomba_Clean,ON)
        }
    }
}
end

```