

## Turtlebot von 0 auf Fahren.

1. Private Hotspot einschalten in Laptop oder Handy
2. Turtlebot über HDMI mit einem Bildschirm verbinden, Tastatur über USB anschließen.

3. Raspberry Pi einschalten

4. Einloggen Zugangsdaten:

User: ubuntu

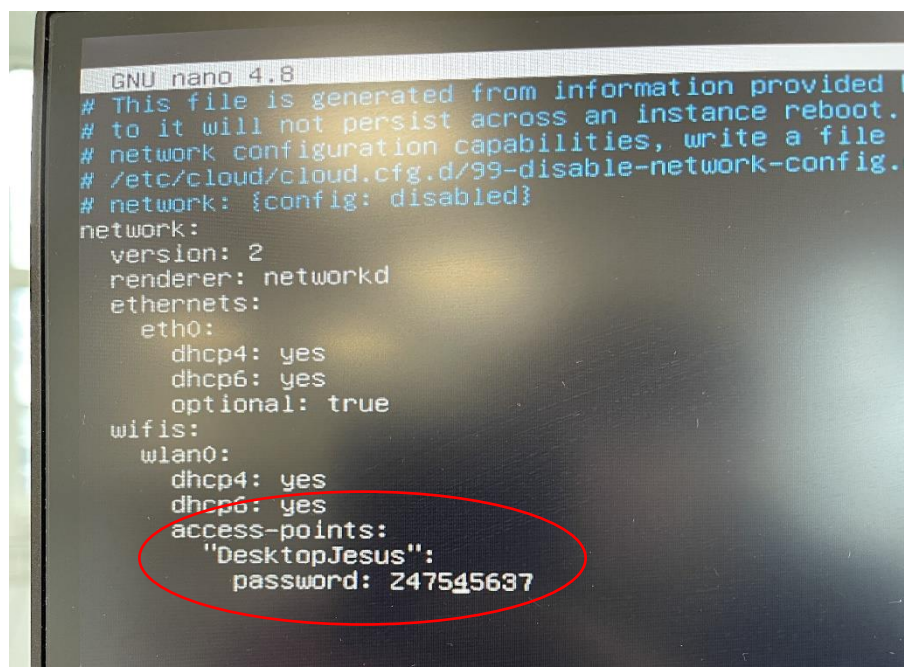
Password: turtlebot

5. Folgende Befehle in Terminal eingeben

```
$ cd etc/netplan
```

```
$ sudo nano 50-cloud-init.yaml
```

In dem File unter "access-points" die neuen Wifi SSID und Password eingeben.



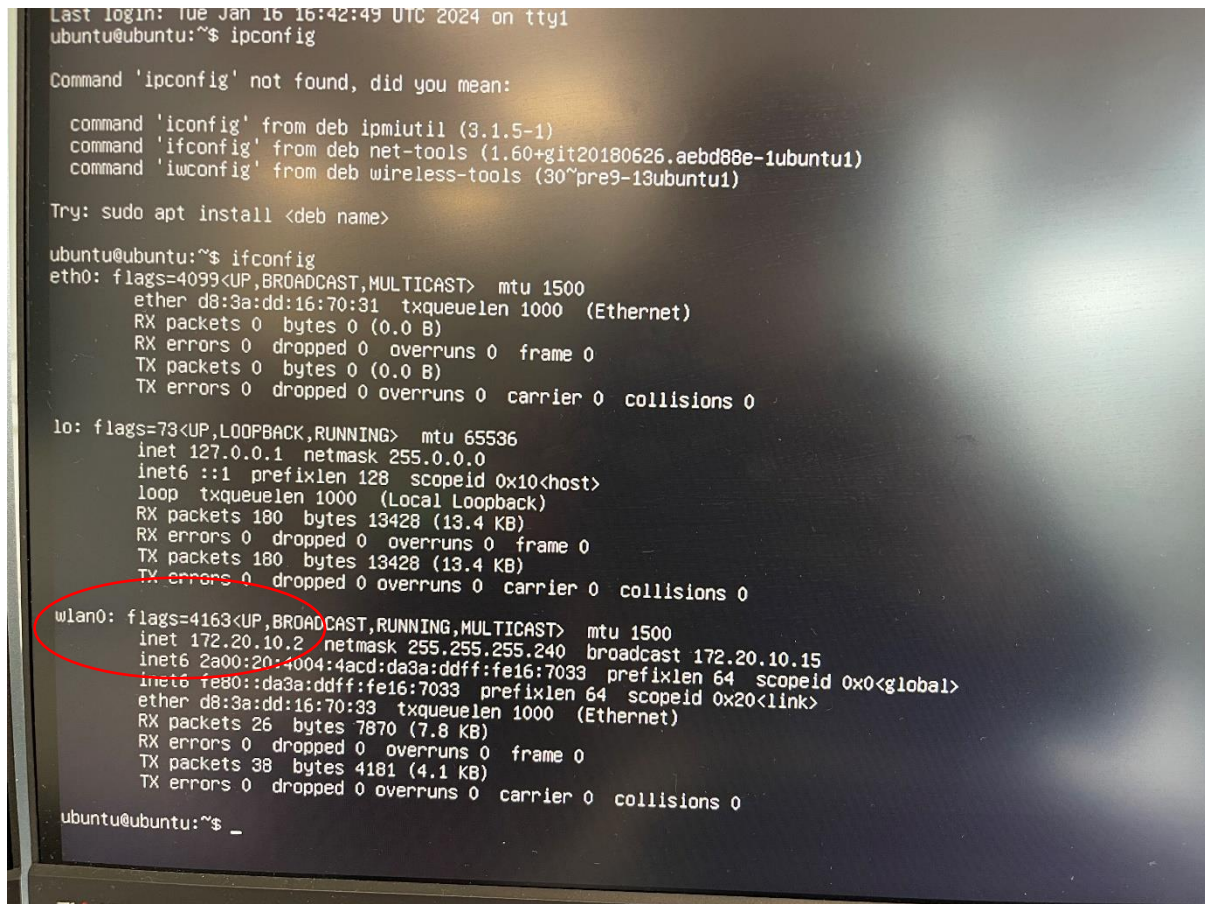
6. Für speichern Strg+X und Y drücken und folgender Befehl eingeben

```
$ sudo reboot
```

7. Der Raspberry Pi soll sich mit dem Internet verbinden

8. Die IP-Adresse ist erforderlich. Um die IP-Adresse zu bestimmen, nutzen wir das folgende Befehl.

\$ ip config



```
Last login: Tue Jan 16 16:42:49 UTC 2024 on tty1
ubuntu@ubuntu:~$ ipconfig

Command 'ipconfig' not found, did you mean:

  command 'iconfig' from deb ipmiutil (3.1.5-1)
  command 'ifconfig' from deb net-tools (1.60+git20180626.aebd88e-1ubuntu1)
  command 'iwconfig' from deb wireless-tools (30~pre9-13ubuntu1)

Try: sudo apt install <deb name>

ubuntu@ubuntu:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether d8:3a:dd:16:70:31 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
      loop txqueuelen 1000 (Local Loopback)
      RX packets 180 bytes 13428 (13.4 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 180 bytes 13428 (13.4 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 172.20.10.2 netmask 255.255.255.240 broadcast 172.20.10.15
      inet6 2a00:20:4004:4acd:da3a:ddff:fe16:7033 prefixlen 64 scopeid 0x0<global>
      inet6 fe80::da3a:ddff:fe16:7033 prefixlen 64 scopeid 0x20<link>
      ether d8:3a:dd:16:70:33 txqueuelen 1000 (Ethernet)
      RX packets 26 bytes 7870 (7.8 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 38 bytes 4181 (4.1 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ubuntu:~$ _
```

Unter wlan0: steht die IP-Adresse unter inet.

Mit der IP-Adresse kann man jetzt den Turtlebot von HDMI und Tastatur trennen und weiter über das Remote-PC mit SSH steuern.

9. Das Remote-PC mit der Privater Hotspot verbinden und in einem Tilix Terminal folgender Befehl eingeben.

Ssh ubuntu@"IP-Adresse des Raspberry Pi"

Die Anweisung zustimmt und mit den Zugangsdaten des Turtlebots sich einloggen.

10. Die IP-Adresse des Remotes-PC ist auch erforderlich. In eine neue Tilix Terminal folgender Befehl eingeben.

\$ ifconfig

Läuft gleich wie Punkt 8.

11. Die IP-Adressen muss in den erforderlichen Dateien eingeben. Dafür gehen wir in der Remote-PC mit den folgenden Befehle.

```
$ nano ~/.bashrc
```

In dieser Datei ändern wir die:

ROS\_Master\_URI und ROS\_HOST\_NAME mit der IP von Remote-PC.

Mit Strg+X speichern und die Änderungen Sourcen mit dem Befehl

```
$ source ~/.bashrc
```

12. In der Raspberry Pi über SSH machen wir die gleiche Reihenfolge aber in ROS\_HOST\_NAME geben wir die IP-Adresse von der Raspberry Pi.

Damit ist die WLAN-Kommunikation zwischen Remote-PC und Turtlebot abgeschlossen.

Turtlebot Bringup

1. Für das Bringup des Turtlebots folgende Befehle über SSH im Turtlebot eingeben.

```
$ export TURTLEBOT3_MODEL=$burger
```

```
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

Mit diesem Schritt der Turtlebot ist Betriebsbereit und wartet auf Einweisungen von ROS.

2. Im Remote-PC ros core starten mit dem Befehl

```
$ roscore
```

Dieses Terminal muss immer laufen.

3. Folgende Befehle müssen **\*IMMER\*** eingegeben werden in jeden neuen Terminal, wo man Ros Packages laufen will.

```
$ cd Dokumente/catkin_ws-main/
```

```
$ catkin_make
```

```
$ source ~/Dokumente/catkin_ws-main/devel/setup.bash
```

4. Jetzt kann man die ROS Nodes starten. Jede Node muss in einem neuen Terminal gestartet werden.

1. Pylon Kamera mit dem Befehl

```
$roslaunch pylon_camera pylon_camera_node.launch
```

## 2. HSV Param

```
$roslaunch turtlebot_test hsv_param.py
```

Mit diesem Node werden die Farben der LED-Matrizen gefiltert um ihre HSV-Werten zu bestimmen. Die HSV-Werten müssen in der detect\_turtlebot.py Datei eingegeben werden.

## 3. Detectturtlebot

```
$ roslaunch turtlebot_test detect_turtlebot.py
```

## 4. Pfadfolge node

```
$ roslaunch ros_package ros_package_copy.py
```