




Openauto

How to set up the initial image of Open Auto

- Install OpenAuto on microSD
- Insert microSD into RPI ZERO 2W and boot
- After powering off the RPI, disconnect the microSD and connect it to the external memory of the PC.
- Download the boot folder from  [PI-SIGHT_SW_Openauto](#)
- Copy the config.txt file from the downloaded boot folder and paste it into the boot directory of the microSD.

- Modify and copy the crankshaft_env.sh file in the downloaded /boot/crankshaft folder and paste it into the /boot/crankshaft directory of the microSD (modify it with your network SSID and PW)

```
/boot/crankshaft/crankshaft_env.sh # Wifi client mode (use this entry
for home/work wifi - not for phone's hotspot) # If your SSID or
password contains special chars or spaces make sure using quotation
marks ="SSID" / ="password" WIFI_SSID="PIPA" WIFI_PSK="qwertyuiop" #
Hotspot (if enabled the wifi client is disabled and a hotspot is
opened) # Hotspot has now a default password (1234567890) -> changeable
in /etc/hostapd/hostapd.conf if really needed! ENABLE_HOTSPOT=0
```

- Copy the config.txt file from the downloaded boot folder and paste it into the boot directory of the microSD.
- Copy openauto.ini and hotspot.conf files from the /boot/crankshaft folder and paste them into the /boot/crankshaft directory on the microSD.
- Disconnect the microSD from the PC, connect it to the RPI and boot it.
- RPI SSH connection from PC
- sudo raspi-config → camera on
- Edit file

```
#Edit content nano /etc/resolv.conf nameserver 8.8.8.8 #Edit content
nano /etc/fstab /boot vfat defaults,noatime,nodiratime 0 2 #Add content
nano /etc/rc.local sh /boot/crankshaft/bootrun.sh
```

- Installing essential components

```
sudo apt-get update sudo apt-get upgrade sudo pip install --upgrade pip
sudo pip3 install picamera keyboard pyautogui
```

- Setting display output related to pyautogui

```
sudo apt-get install python3-tk python3-dev sudo pip3 install python-
xlib sudo echo "DISPLAY=:0" >> ~/.bashrc sudo source ~/.bashrc
```

- Bluetooth controller connection settings

```
sudo apt-get install bluetooth blueman bluez bluez-cups libbluetooth-
dev pi-bluetooth python-bluez sudo sed -i 's/Privacy = off/# Privacy =
off/' /etc/bluetooth/main.conf sudo sed -i 's/ControllerMode =
bredr/#ControllerMode = dual/' /etc/bluetooth/main.conf sudo sed -i
's/FastConnectable = true/#FastConnectable = false/'
/etc/bluetooth/main.conf sudo systemctl disable tap2wake sudo systemctl
disable csng-bluetooth sudo systemctl disable btautopair sudo systemctl
disable btdevicedetect sudo systemctl disable btrestore sudo systemctl
disable btservice sudo systemctl enable bluetooth sudo systemctl enable
hciuart sudo systemctl enable ofono sudo sed -i 's/load-module module-
bluetooth-discover headset=ofono/load-module module-bluetooth-discover
autodetect_mtu=yes/' /etc/pulse/default.pa sudo sed -i 's/load-module
module-bluetooth-discover headset=ofono/load-module module-bluetooth-
discover autodetect_mtu=yes/' /etc/pulse/system.pa sudo nano /etc/fstab
-> /tmp/bluetooth 줄 삭제 sudo rm -r /var/lib/bluetooth sudo mkdir
/var/lib/bluetooth sudo chmod -R 777 /var/lib/bluetooth sudo chown -R
pi /var/lib/bluetooth
```

- Pairing remote control with bluetoothctl manual command, trust test
- Reboot

- Check if the remote control is properly saved in paired-devices with the bluetoothctl manual command.

- Disable unnecessary services

```
sudo systemctl disable gpio2kbd sudo systemctl disable
usbdetect.service sudo systemctl disable usbrestore.service sudo
systemctl disable triggerhappy.service sudo systemctl disable
triggerhappy.socket sudo systemctl disable gpsd
```

- Create LED off system service

```
sudo nano /etc/systemd/system/disable-led.service [Unit]
Description=Disables the LED After=multi-user.target [Service]
Type=oneshot RemainAfterExit=yes ExecStart=sh -c "echo 0 | sudo tee
/sys/class/leds/led0/brightness > /dev/null" ExecStop=sh -c "echo 1 |
sudo tee /sys/class/leds/led0/brightness > /dev/null" [Install]
WantedBy=multi-user.target sudo systemctl enable disable-led.service
```

- LED off system service running

```
sudo systemctl enable disable-led.service
```

- Edit file

```
#Edit content /etc/resolv.conf nameserver 8.8.8.8
```

- I2S MIC Settings

```
sudo pip3 install --upgrade adafruit-python-shell sudo wget
https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-
Scripts/master/i2smic.py sudo python3 i2smic.py Install after checking
RPI ZERO 2W device loading in i2smic.py script
```

- Power off RPI

- After detaching the microSD, connect it as external memory to another Linux device.

- Working with microSD GParted on other Linux devices (extending boot partition)

Resizing the Raspberry Pi Boot Partition

Though the Raspberry Pi computer is eminently networkable, some projects still just work best by physically moving the SD

★ <https://learn.adafruit.com/resizing-raspberry-pi-boot-partit...>

<div>/dev/sda1 5.26 GiB</div>			<div>/dev/sda2 1.95 GiB</div>		
Partition	File System	Mount Point	Label	Size	Used
/dev/sda1	fat32	/media/pi/boot	boot	5.26 GiB	10.53
/dev/sda2	ext4	/media/pi/19978116-b968-4738-93ef-4d1af6861b99		1.95 GiB	1014.60

- After GParted operation, connect the microSD as external memory to your PC.
- After GParted is done, copy all the files from the downloaded /boot/crankshaft folder and paste them into the /boot/crankshaft directory on your microSD.
- After creating the /boot/Video Converter directory, paste My MP4Box GUI.zip
- Disconnect the microSD from the PC, connect it to the RPI and boot it.
- Tested all functions including remote control pairing, remote control key input, Android Auto operation, microphone input, camera operation, etc.
- After powering off, remove the microSD and connect it to the PC's external memory.

te d'origine
luez cette traduction
e avis nous aidera à améliorer Google Traduction