

Raspberry Pi Setup

Raspberry Pi

Raspberry Pi is a single board computer the size of a credit card that can run a full featured Linux operating system. [Version 3 model B+](#) features a quad core ARM CPU clocked at 1.4 GHz, strong enough for real time image processing. The board has a dedicated connector for the [camera module](#), and if needed, additional cameras can be connected through the 4 available USB ports. Moreover, the board contains a wireless module that supports Bluetooth, Wi-Fi 2.4 GHz, and Wi-Fi 5 GHz, which can be used for remote connection and development.

Since the board will be mounted on a 1/10 scale car model and will operate headless (without a monitor), we recommend that you install the [Raspbian Stretch Lite](#) Linux distribution which comes without a graphical interface installed. However, you can still access the command line interface if you connect a monitor to the HDMI port. There are two advantages of running without a graphical interface. First, your program will not be interrupted by the many processes spawned by the graphical interface; eliminating this makes your program more predictable which is a desired feature for real-time control. Secondly, the sounds produced by the graphical interface would interfere with the output of the control algorithm, and we will need the two audio timers of the left and the right headphones to control the car speed and steering angle.

Moreover, the board has a 40-pin connector with the following [pinout](#) that exposes additional hardware features such as UART, I2C and SPI protocols, PWM output signals, configurable input/output ports, and power.

Attention

If you received a preconfigured SD card skip the next two sections and continue with "Remote connection" section!

Flashing the SD card

We will install Raspbian, a Linux distribution based on Debian designed for Raspberry Pi. For that, download the latest Raspbian Lite image from the [official website](#) and extract the image from the zip file. Next, download [Etcher](#), a special program that can copy this type of image files on the SD card. Note, the plain copy-paste of the image on the SD card will not work in this case as the data on the SD card should follow a specific format in order for the board to boot the Raspbian. In Etcher, first select the image then the SD card drive and press flash to correctly transfer the image on the SD card. Once the process completes, the SD card will have two partitions: boot and rootfs. You will see both partitions if your operating system is Linux, otherwise, just the boot one.

Reference: [Image installation – official documentation page](#)

Before booting the freshly install system, a few additional changes must be made.

First, and most importantly, we have to enable the `ssh` ([secure shell](#)) server that will allow us to access remotely the system. For that, open a plain text editor (Notepad on Windows, TextEdit on MacOS, and Gedit on Linux) and save an empty file on the SD card boot partition under the name `ssh`. Make sure the file has no extension such as `.txt`. During the boot process, Raspbian checks if the boot partition contains a file named `ssh`, and if so, starts the `ssh` server.

Secondly, to access to the Raspberry Pi through `ssh`, both your computer and the board must be on the same network. For that, we have to create a wireless configuration file with the network details. Similar to first step, open a plain text editor (Notepad on Windows, TextEdit on MacOS, and Gedit on Linux) and create a new file with the following content:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
    ssid="Race On 5G"
    psk="Fight On"
    priority=1
    id_str="workshop-router-5G"
}

network={
    ssid="Race On 2.4G"
    psk="Fight On"
    priority=2
    id_str="workshop-router-2.4G"
}
```

These are the connection details for the two networks of the Race On router. To add you own router, append at the end the file the following text changing the the `ssid` and `psk` to match your network name and password:

```
network={
    ssid="Network-Name"
    psk="Network-Password"
    priority=3
    id_str="home-router"
}
```

If your network does not have a password, replace the `psk` line with `key_mgmt=NONE`. Now save the file under the name `wpa_supplicant.conf` on the SD card boot partition. This file will be automatically moved during the boot process to its correct location `/etc/wpa_supplicant/wpa_supplicant.conf`. However, this will replace your previously configured wireless networks. Therefore, when you want to add a new network, first copy the previous configuration and then append the details of the network you want to add.

And lastly, enable the `PWM` ([Pulse Width Modulation](#)) timers used to control the motor speed and the servo angle. Open `config.txt` from the boot partition using a plain text editor and add at the end of the file the following text:

```
# Enable hardware PWM on pin 12 and 13
dtoverlay=pwn-2chan,pin=12,func=4,pin2=13,func2=4
```

This enables the hardware timer that will generate the pulses on pins 12 and 13 that control the motor power and the servo angle. Save the file, then safely remove the card and insert into the Raspberry Pi board.

Reference:

- [ssh setup](#)
- [wireless setup documentation](#)
- [config.txt documentation](#)

Power the board and wait for the green light (CPU activity) to turn off that will indicate the end of the booting process. If configured right, the Raspberry Pi board now should be ready and waiting for remote connections. To access the board, connect your computer to the same network as the one configured in `wpa_supplicant.conf`, i.e., Race On 5G. Then run in terminal the following command: `ssh pi@<hostname>.local`, where `hostname` is `raspberrypi` if the SD card was manually configured or `raspberrypi-ID` if you received a preconfigured SD card and ID is your card number. Note, Windows does not come with an ssh client installed, Mac OS and Linux do, and therefore you need to download PuTTY, and ssh client for Windows.

If everything went right, you should see a request to accept a certificate. Type `yes` to accept and for the password use `raspberry`. Now as you accessed the board remotely change the default password using the `passwd` command and follow the instructions.

Bonus – First Selfie

To take the first selfie using the Raspberry Pi camera module, first, enable the camera module using the raspberry pi configuration tool `sudo raspi-config` and select `Interface Tools` then `Enable camera`. Save and accept to reboot the board. Connect again using the ssh and take your first pi-selfie using the command `raspistill -o image.jpg`. Type `exit` to disconnect the ssh session and `scp pi@<hostname>.local:image.jpg .` to download the image to your computer. Use `pwd` command to find the directory where you downloaded the image on your computer.

Feedback

If something is unclear or you found a mistake, please email vbalaban@usc.edu.

Contact persons:
Valeriu Balaban <vbalaban@usc.edu>
Kwame Wright <kwamelaw@usc.edu>

Race On! – Self-Driving Car Competition
Event enable by the [Ming Hsieh Institute](#)

Ming Hsieh Institute
Ming Hsieh Department of Electrical Engineering