

IEEE CCWC 2021 IoT Workshop

Windows Getting Raspberry Pi Images

2021-01-06

Phillip G. Bradford¹

SUMMARY

This document shows how to download a Raspberry Pi image. It also shows how to run a Raspberry Pi image using qemu-system-arm. This is for Windows 10.

DETAILS

Each step in Part 2 works to get images of Raspberry Pis working on your machine. These steps discuss some technical details of virtual Raspberry Pis using QEMU. These steps should be completed at least one day before the workshop. All these installations and downloads are at your own risk.

STEPS

Step 1 requires git to be installed. If you do not have it installed already, then please see the accompanying document: **[Part 2 Windows Install GIT.pdf](#)**

¹ University of Connecticut, Stamford, CT
USA, phillip.bradford@uconn.edu, phillip.g.bradford@gmail.com

STEP 1: CREATE FOLDERS ON YOUR WINDOWS MACHINE

On Windows these folders can be created using the **File Explorer** or a **Command Prompt (Terminal)**.

Our objective is to build the following folders

Folder	Commands
C:\users\your-id\workshop	cd . ; mkdir workshop
~/workshop/RPi1	git clone https://github.com/wonder-phil/RPi
~/workshop/RPi2	After images are in RPi1 then duplicate RPi1 to RPi2
~/workshop/code	git clone

Assuming your-id is your login-id on your Windows 10 machine, then here are the Terminal commands to create the folders:

```
Windows> cd
```

```
Windows> mkdir workshop
```

```
Windows> cd workshop
```

```
Windows> git clone https://github.com/wonder-phil/RPi
```

```
Windows> dir
```

```
3.2M Jul 14 12:40 gemukernel
```

```
7.9K Jul 14 12:39 versatile-pb.dtb
```

These two files are in your workshop/RPi folder
The gemukernel file is the Raspberrian operating system.

The versatile-pb.dtb is a mother board emulated by QEMU. Technically these are ARM Versatile/PB (ARM926EJ-S). Others are available as well.

STEP 2: DOWNLOAD RASPBERRY PI IMG

<https://www.raspberrypi.org/software/operating-systems/>

Download Paspberry Pi OS Lite (438MB), 2-December-2020

Download **the Raspberry Pi OS Lite** image as shown below:

Raspberry Pi OS with desktop and recommended software

Release date: December 2nd 2020

Kernel version: 5.4

Size: 2,949MB

[Show SHA256 file integrity hash:](#)

[Release notes](#)

Download

[Download torrent](#)

Raspberry Pi OS with desktop

Release date: December 2nd 2020

Kernel version: 5.4

Size: 1,177MB

[Show SHA256 file integrity hash:](#)

[Release notes](#)

Download

[Download torrent](#)

Raspberry Pi OS Lite

Release date: December 2nd 2020

Kernel version: 5.4

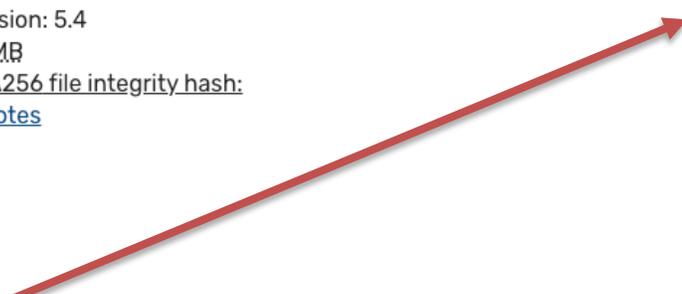
Size: 438MB

[Show SHA256 file integrity hash:](#)

[Release notes](#)

Download

[Download torrent](#)



This comes down as a zip folder containing a single file. Unzip this folder getting the file:

2020-12-02-raspbios-buster-armhf-lite.img

Place this file in ~/workshop/RPi

Windows> cd ~/workshop/RPi; ls -lh

```
1.7G Dec  2 07:56 2020-12-02-raspbios-buster-armhf-lite.img
3.2M Jul 14 12:40 qemukernel
7.9K Jul 14 12:39 versatile-pb.dtb
```

STEP 3: RUN A RASPBERRY PI QEMU VIRTUAL MACHINE

Windows> cd ~/workshop/RPi

Check the IMG file:

Windows> qemu-img info 2020-12-02-raspbios-buster-armhf-lite.img

```
image: 2020-12-02-raspbios-buster-armhf-lite.img
file format: raw
virtual size: 1.73 GiB (1858076672 bytes)
disk size: 1.73 GiB
```

Now convert the IMG file to a QCOW2 file:

```
Windows> qemu-img convert -f raw -O qcow2 2020-12-02-raspbios-buster-armhf-lite.img 2020-12.qcow2
```

```
Windows> qemu-img resize 2020-12.qcow2 +2G
```

```
Windows> qemu-system-arm -M versatilepb \  
-cpu arm1176 \  
-m 256 -hda "./2020-12.qcow2" \  
-net nic -net user,hostfwd=tcp::5022-:22 \  
-dtb "./versatile-pb.dtb" \  
-kernel "./qemukernel" \  
-append "root=/dev/sda2 panic=1 rootfstype=ext4 rw" -no-reboot
```

Then a new screen should pop up showing the Raspberry Pi instance booting:

```
QEMU
[raspberrypi logo]
n for the first loaded unit using IP firewalling.)
random: systemd: uninitialized urandom read (16 bytes read)
random: systemd: uninitialized urandom read (16 bytes read)
systemd[1]: Listening on udev Kernel Socket.
random: systemd: uninitialized urandom read (16 bytes read)
systemd[1]: Listening on initctl Compatibility Named Pipe.
systemd[1]: Started Dispatch Password Requests to Console Directory Watch.
systemd[1]: Condition check resulted in Journal Audit Socket being skipped.
systemd[1]: Condition check resulted in Arbitrary Executable File Formats File S
system Automount Point being skipped.
systemd[1]: Condition check resulted in Kernel Module supporting RPCSEC_GSS bein
g skipped.
systemd[1]: Created slice system-systemd\x2dfsck.slice.
systemd[1]: Reached target Swap.
systemd[1]: Starting Restore / save the current clock...
systemd[1]: Listening on udev Control Socket.
systemd[1]: Starting udev Coldplug all Devices...
systemd[1]: Mounting RPC Pipe File System...
systemd[1]: Mounted POSIX Message Queue File System.
systemd[1]: Started Load Kernel Modules.
systemd[1]: Mounted RPC Pipe File System.

Raspbian GNU/Linux 10 raspberrypi tty1
raspberrypi login:
```

raspberrypi login: pi
raspberrypi password: raspberry

If your mouse cursor gets lost, then:



To shut
shut it down

Raspberrypi> sudo shutdown -h now

EXPLANATION OF THE QEMU-SYSTEM-ARM COMMAND

qemu-system-arm is the executable that we use from the QEMU binaries we downloaded earlier. This is what allows us to spin up the virtual machine.

-M versatilepb This names the machine type for QEMU.

-cpu arm1176 makes QEMU emulate the ARM1176 processor. This is the processor used by Raspberry Pis.

-m 256 sets QEMU to use 256MB memory for the emulated Raspberry Pi.

-hda "./2020-12.qcow2" sets QEMU to use the file image 2020-12.qcow2. Qcow2 is QEMU copy-on-write format.

-net nic -net user,hostfwd=tcp::5022-:22 The **-net nic** indicates there is a network interface card. The **"hostfwd=tcp::5022:22"** maps the host port 5022 to the raspberry pi's port 22 on the virtual machine. This allows ssh access.

-dtb "./versatile-pb.dtb" just maps the file we have saved in our folder to the **-M** flag we specified earlier.

-kernel "./qemukernel" This points QEMU to the Raspberry Pi Raspberrian kernel.

-append "root=/dev/sda2 panic=1 rootfstype=ext4 rw" these specifics are appended on to the boot command.

-no-reboot This tells QEMU to stop when the Raspberry Pi shuts down rather than rebooting on shutdown.