

<https://www.halvorsen.blog>



Raspberry Pi

Hans-Petter Halvorsen

Contents

- What is Raspberry Pi?
- Getting Started with Raspberry Pi
 - What Do You Need?
 - Raspberry Pi Operating System
 - The Linux Terminal
 - Tools and Remote Access
- Raspberry Pi and Python Programming
 - GPIO, GPIO with Python and GPIO Features
- Raspberry Pi with MATLAB
- Raspberry Pi using ThingSpeak and Python



What is Raspberry Pi?

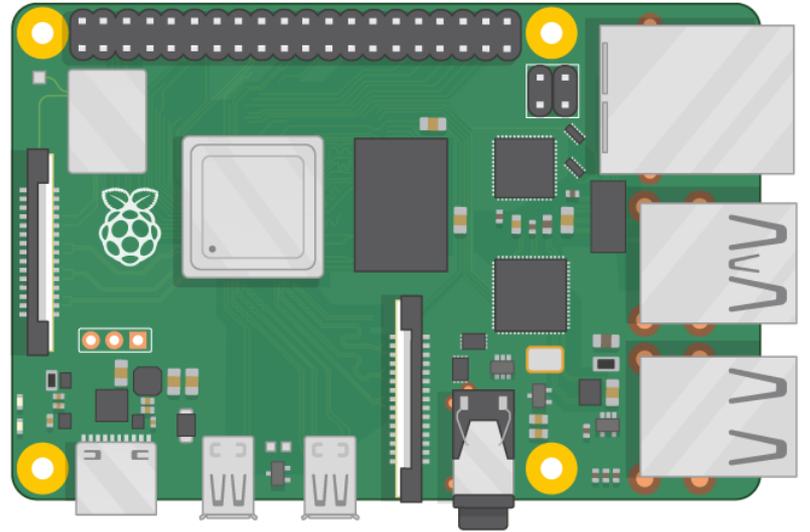
Hans-Petter Halvorsen

[Table of Contents](#)

Raspberry Pi

Raspberry Pi is a **tiny** (about 9x6cm), **low-cost** (\$35+), **single-board computer** that supports embedded **Linux** operating systems

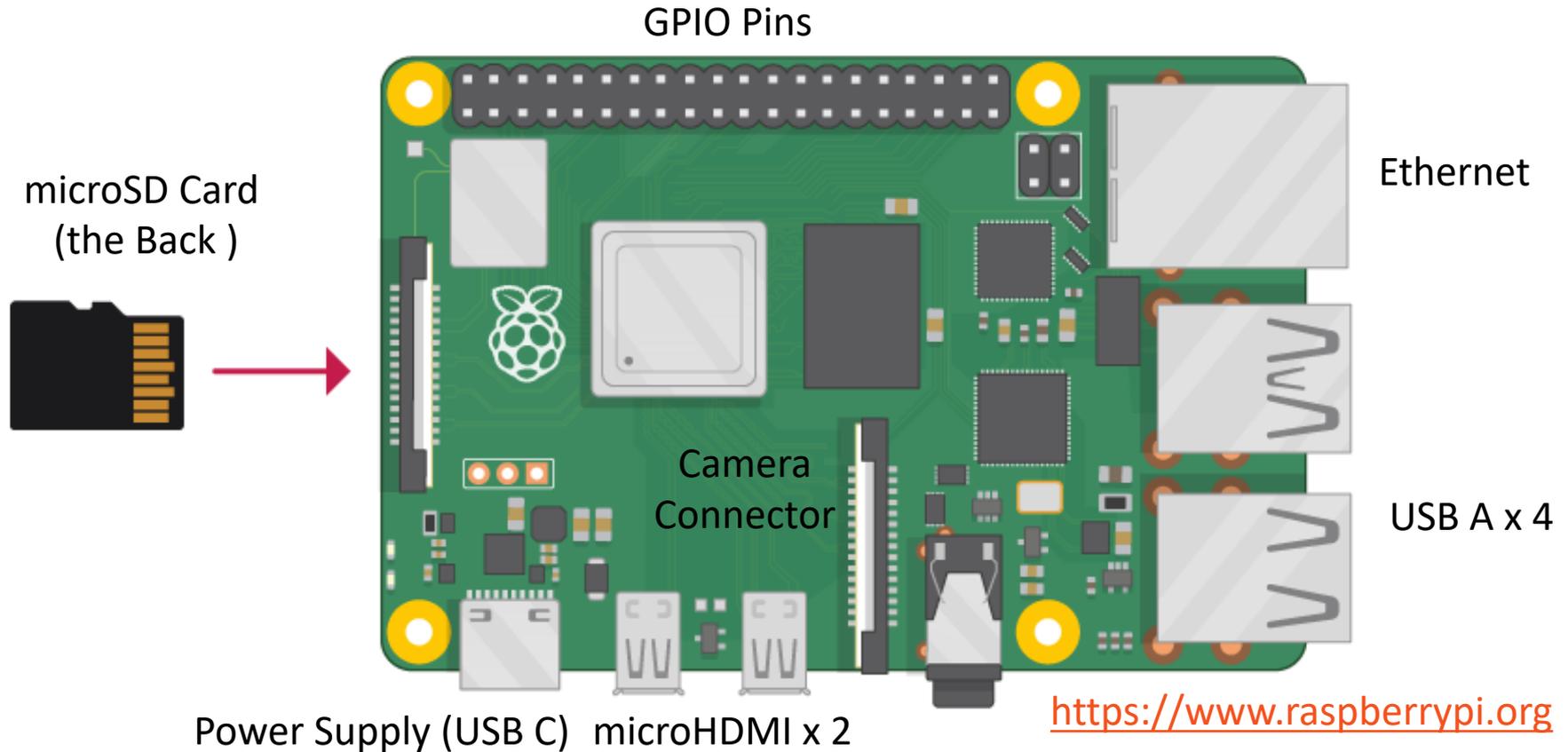
The recommended Operating System is called **Raspberry Pi OS** (Linux based)



Raspberry Pi

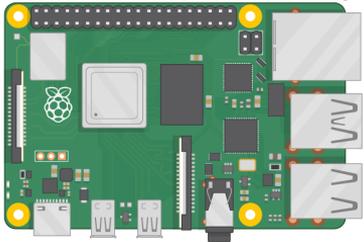
- The Raspberry Pi is a small computer that can do lots of things
- It has a **small footprint (about 9x6cm)** and it is **cheap (\$35+)**
- You plug it into a monitor and attach a keyboard and mouse
- It has so-called **GPIO** pins (General Purpose Input/Output) for connection sensors and other electronic components like LEDs, etc.
- Raspberry Pi is as well suited for **prototyping, datalogging** and different electronics projects, a media center, etc.
- It can be used to learn programming, IT and other technical skills, etc.
- RP has limited power (CPU, RAM, etc.) so it cannot normally replace a desktop computer or laptop for ordinary use
- The “Harddrive” is a Micro SD card

Raspberry Pi



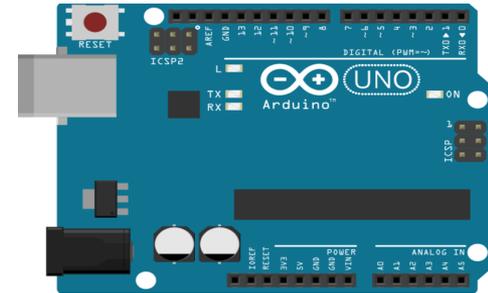
Raspberry Pi vs. Arduino

- Raspberry Pi (RP) is a Microcomputer
- It has an ordinary Operating System (OS)
- You can connect USB devices, Keyboard, Mouse, Monitors, etc.
- It has a “hard-drive“ in form of a microSD card
- RP has Bluetooth, Wi-Fi, and Ethernet connection
- RP has basically all the features an ordinary computer has but in a much smaller package
- 2, 4 or 8 Gb RAM
- RP runs Linux applications



Both have Digital Pins
Both have SPI and I2C
Arduino (UNO) has also
Analog Input Pins

- Arduino is a Microcontroller
- Arduino has a Bootloader and not an ordinary operating system
- Arduino is NOT a computer, only a small controller, whose purpose is to control things
- No Bluetooth, Wi-Fi (some models have), and Ethernet (but can be provided as so-called Shields)
- Very little RAM (a few Kb)
- Inexpensive





Getting Started with Raspberry Pi

Hans-Petter Halvorsen

[Table of Contents](#)



What Do You Need?

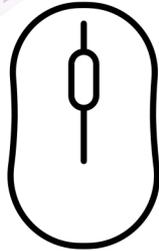
What Do You Need?

- **Raspberry Pi**
- **Micro SD Card (+ Adapter)**
- **Power Supply**
- microHDMI to HDMI Cable
- Monitor
- Mouse
- Keyboard

What Do You Need?

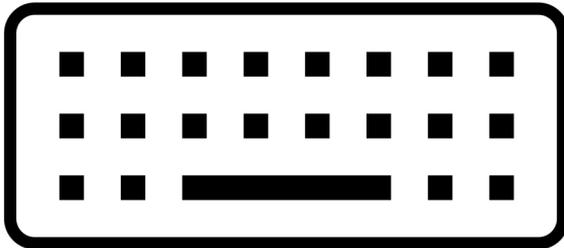


USB C Power Supply
15W (5V, 3A)

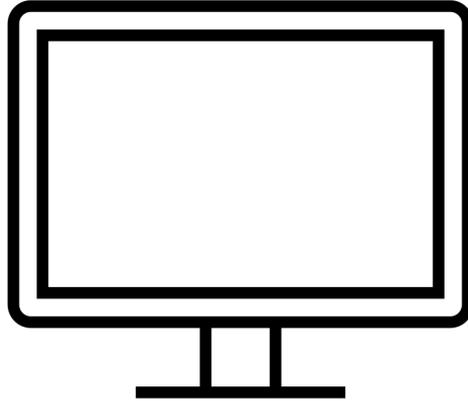


Mouse

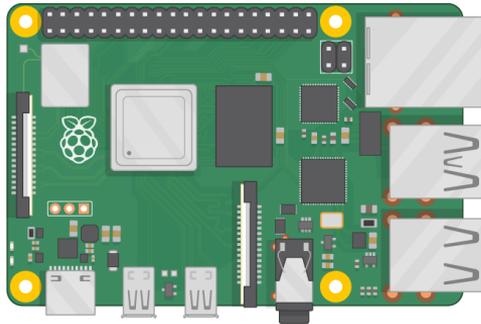
Keyboard



Monitor (with HDMI)



Raspberry Pi



microHDMI to HDMI Cable



Micro SD Card (+ Adapter)

(8GB or more)



<https://www.raspberrypi.org>



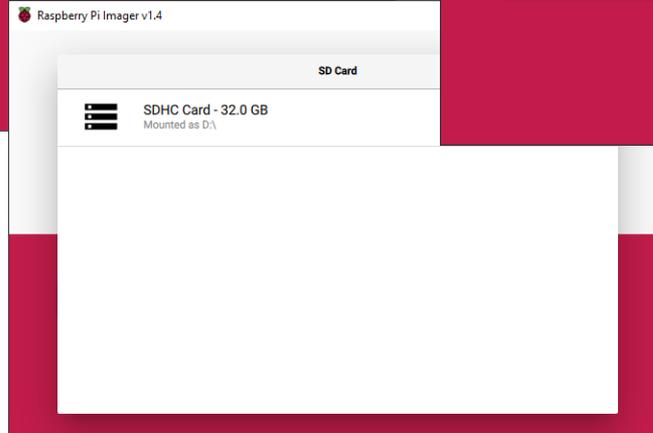
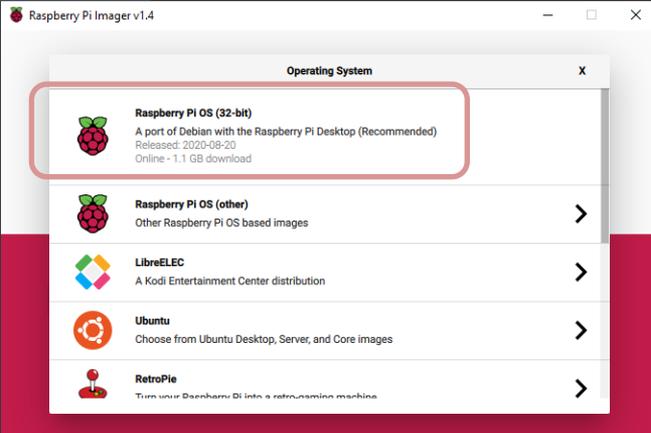
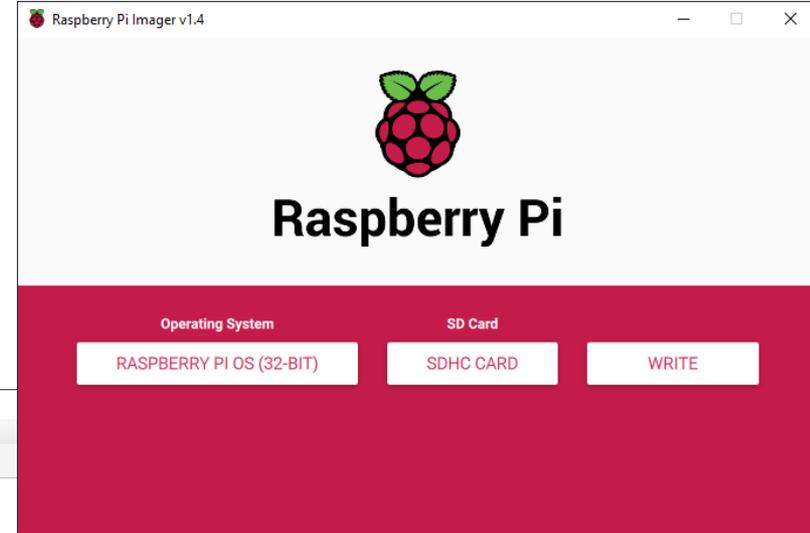
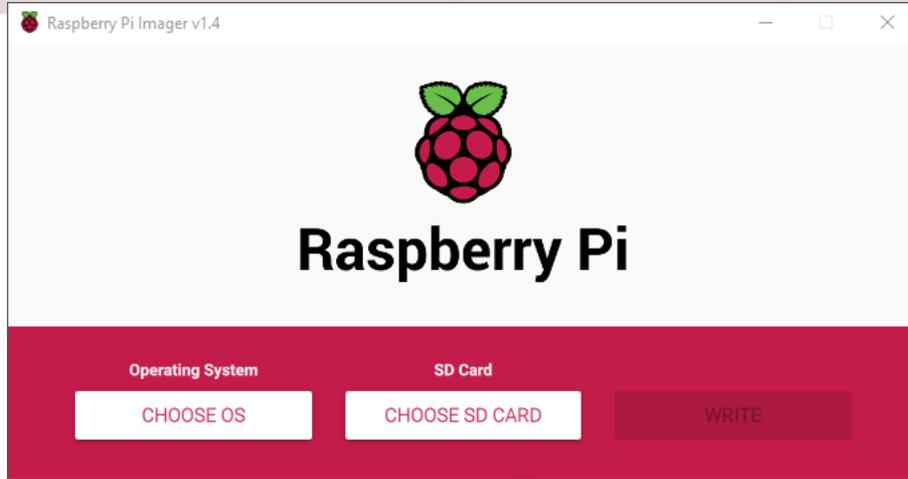
Raspberry Pi OS

Raspberry Pi OS

- In order to make your Raspberry Pi up and running you need to install an Operating System (OS)
- The OS for Raspberry Pi is called **Raspberry Pi OS** (previously known as Raspbian)
- Raspberry Pi runs a version of an operating system called **Linux** (Windows and macOS are other operating systems).
- To install the necessary OS, you need a **microSD** card
- Then you use the **Raspberry Pi Imager** in order to download the OS to the microSD card

<https://www.raspberrypi.org/software/>

Raspberry Pi Imager



Start using Raspberry Pi



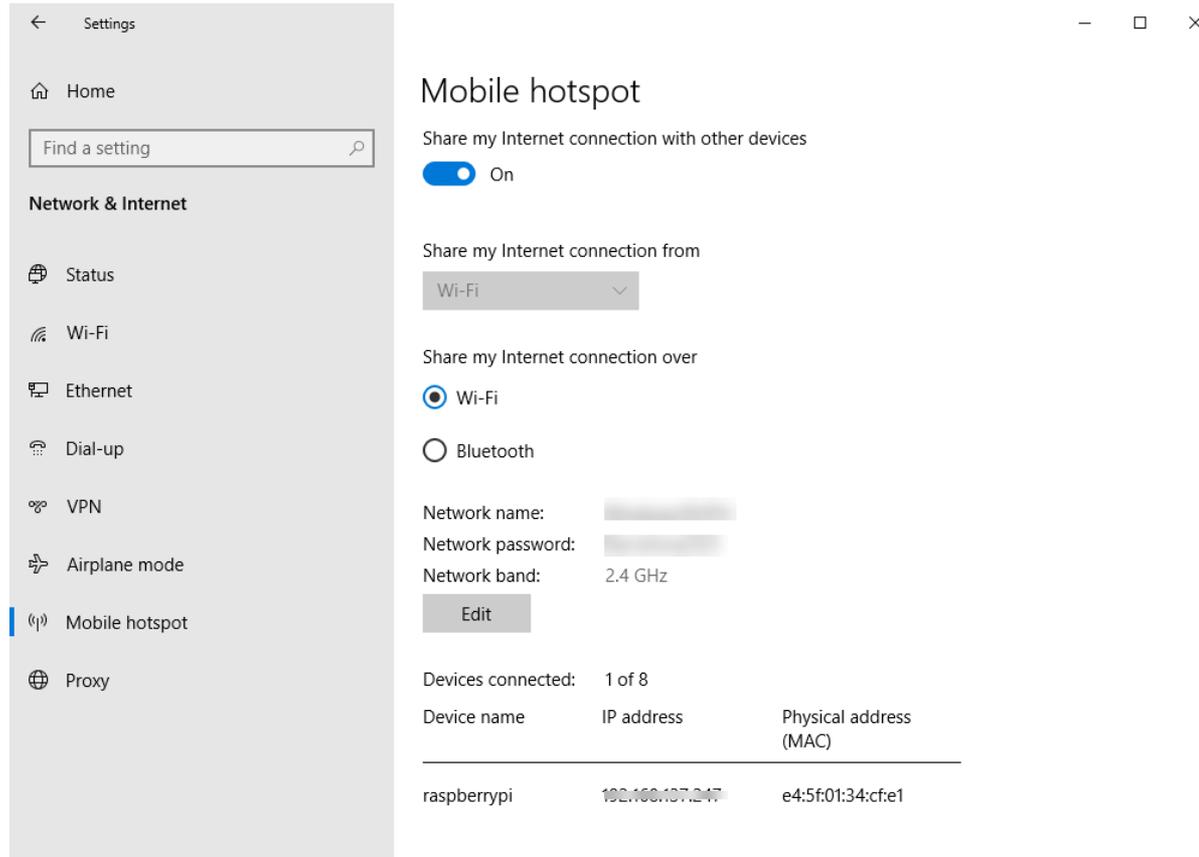
- Put the microSD card into the Raspberry Pi
- Connect Monitor, Mouse and Keyboard
- Connect Power Supply
- Follow the Instructions on Screen to setup Wi-Fi
- etc.

Raspberry Pi OS

Connect to Wi-Fi Network

Don't use the Eduroam WiFi network in your University

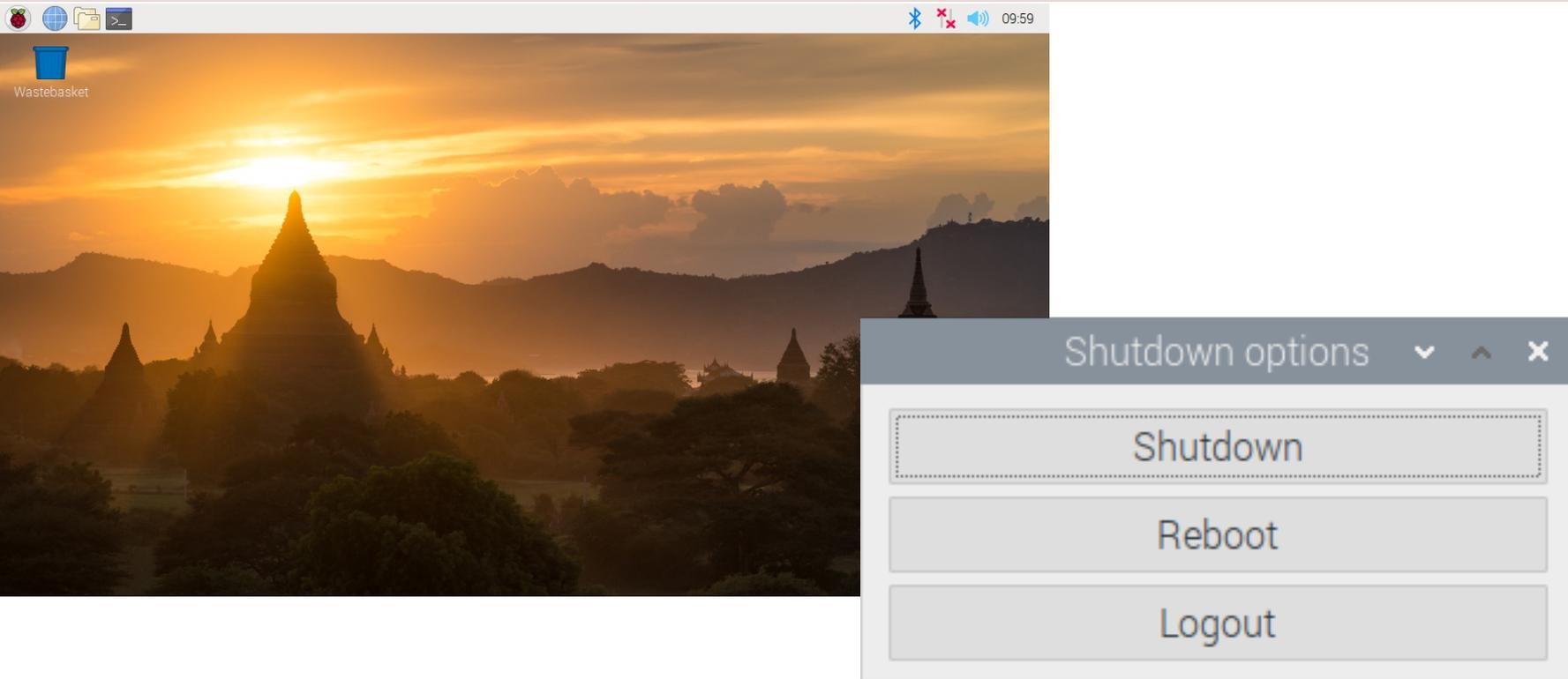
- The first thing you typically need to do is to connect your Raspberry Pi to a Wi-Fi network.
- Your home network normally works fine. If you are in a company or a university, you may get some trouble.
- In these situations, you can configure a Mobile hotspot using your Windows 10/11 PC



The screenshot shows the Windows Settings application. On the left, the 'Settings' sidebar is visible with 'Mobile hotspot' selected. The main pane displays the 'Mobile hotspot' settings. The 'Share my Internet connection with other devices' toggle is turned 'On'. The 'Share my Internet connection from' dropdown is set to 'Wi-Fi'. The 'Share my Internet connection over' radio buttons have 'Wi-Fi' selected. Below these are fields for 'Network name' and 'Network password', both of which are blurred. The 'Network band' is set to '2.4 GHz'. An 'Edit' button is visible. At the bottom, a table shows the status of connected devices.

Device name	IP address	Physical address (MAC)
raspberrypi	192.168.137.247	e4:5f:01:34:cf:e1

Shut-down Raspberry Pi

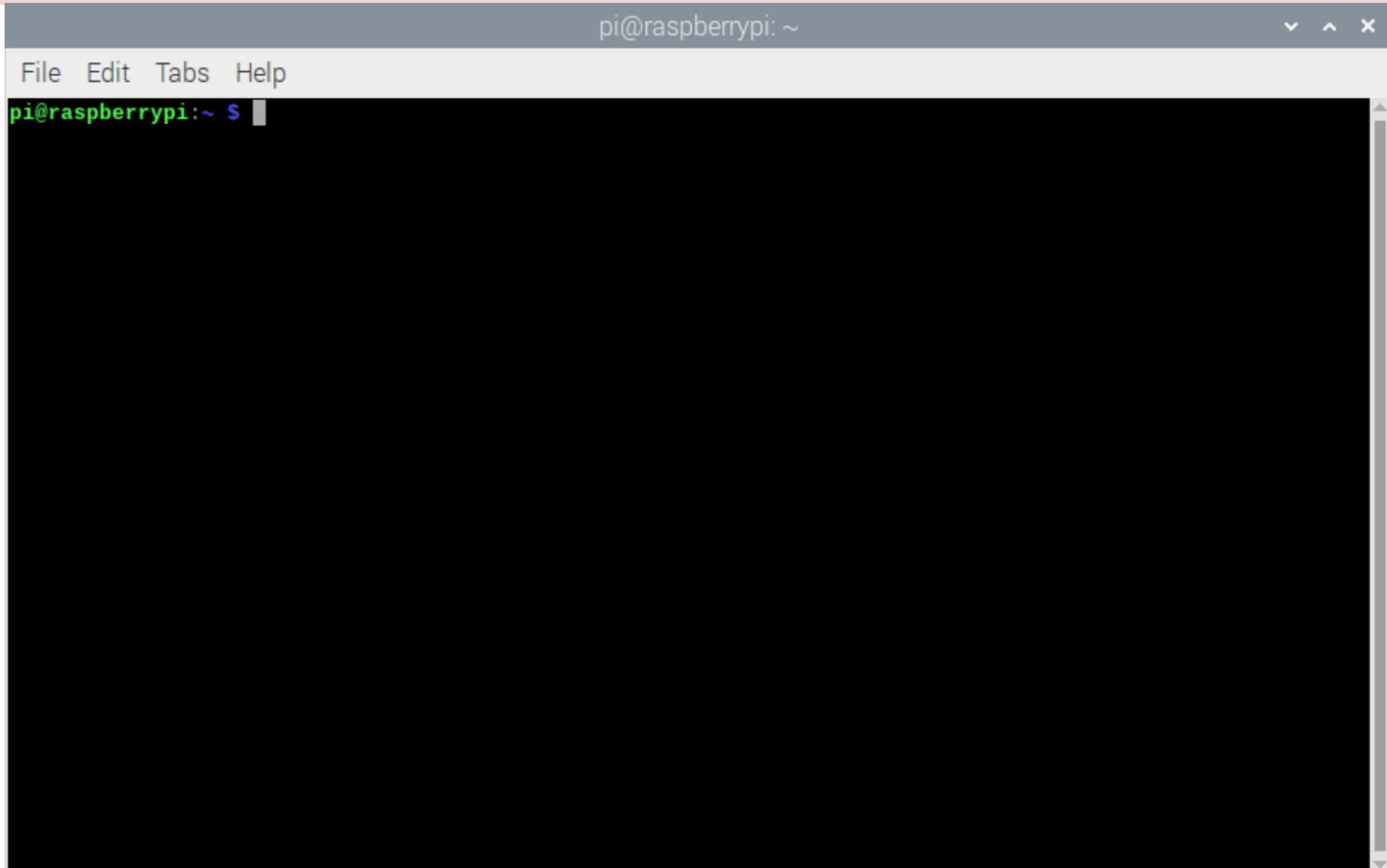


Select the Raspberry Pi icon in upper left corner and select “Logout”



The Linux Terminal

The Terminal



The Terminal

- The Raspberry Pi OS is a Linux based OS and comes with a GUI with limited features, so very often you need to type “low-level” commands using the Terminal
- The Linux terminal is a powerful tool for “low-level” operations on the OS

Update Raspberry Pi OS

- Run the following commands in the Terminal window:
- First run the following:

```
sudo apt update
```

- Then:

```
sudo apt full-upgrade
```

sudo

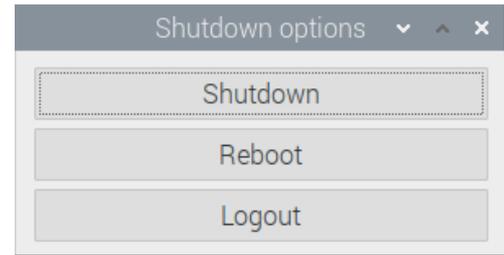
- sudo is a program for Unix-like computer operating systems (like Linux) that allows users to run programs with the security privileges of another user, by default the superuser
- It originally stood for "superuser do"
- Some commands that make permanent changes to the state of your system require you to have root privileges to run.
- The command sudo temporarily gives your account (if you're not already logged in as root) the ability to run these commands
- You typically use it in your Terminal window for installing/upgrading software, etc.

How to Power off your RP

- Unlike your other electronic devices, Raspberry Pi doesn't come with an "off" switch.
- You should not just "pull out the plug"
- Enter the following in the Terminal:

```
sudo poweroff
```

Or use the Graphical Interface:



- This will give you a clean shutdown. After that, you can unplug the computer.
- How to turn on Raspberry Pi after shutdown? It's simple. Just plug it in.

<https://www.proqsolutions.com/shutting-down-your-raspberry-pi/>



Tools

Scrot

- Scrot is a built-in tool for **Screen Capture**
- Scrot is a Command-line tool with no GUI
- Hit “Print Screen” key and the image will be saved in the “/home/pi” folder

Gnome Screenshot

- Gnome Screenshot is better than “Scrot” since it has a desktop GUI
- You need to install it (the Terminal):

```
sudo apt update && sudo apt upgrade  
sudo apt install gnome-screenshot
```
- It will then appear in Menu > Accessories > Screenshot



Remote Access

Remote Access

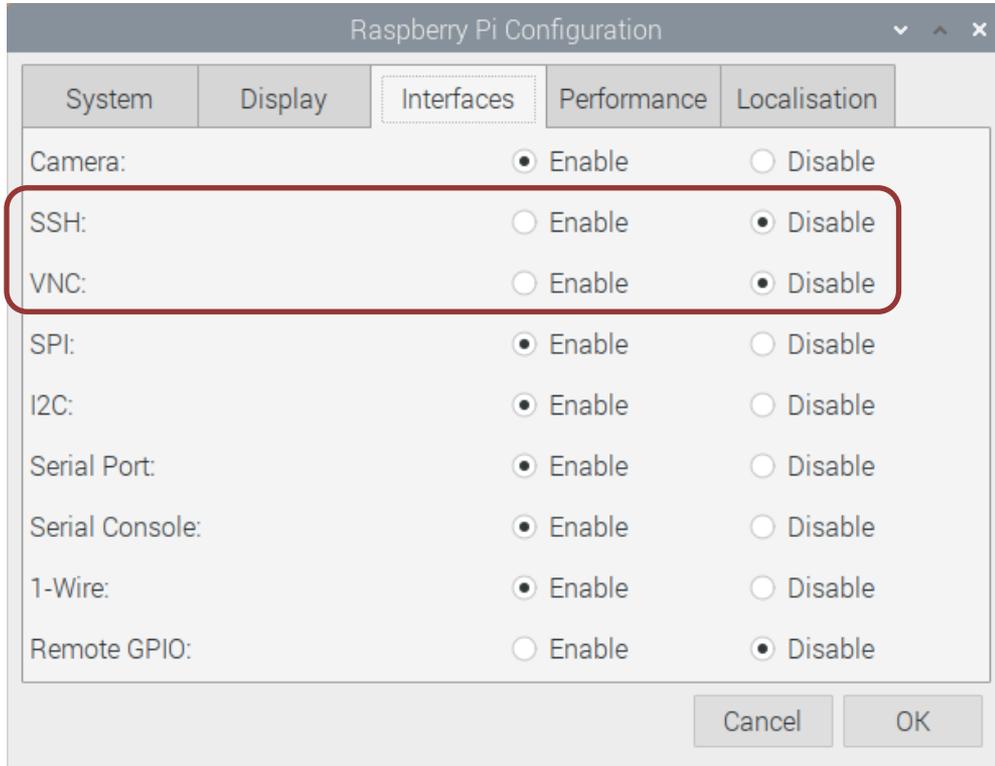
Typically, you want to have Remote Access to you Raspberry Pi from your ordinary Desktop Computer. Different options:

- SSH – Command Line Access

Or use some kind of Remote Desktop software:

- **XRDP/RDC**
- TeamViewer
- VNC

Remote Access



Make sure to Enable the Feature you want to use

Alt1: Remote Desktop

1. Install XRDP

<https://en.wikipedia.org/wiki/Xrdp>

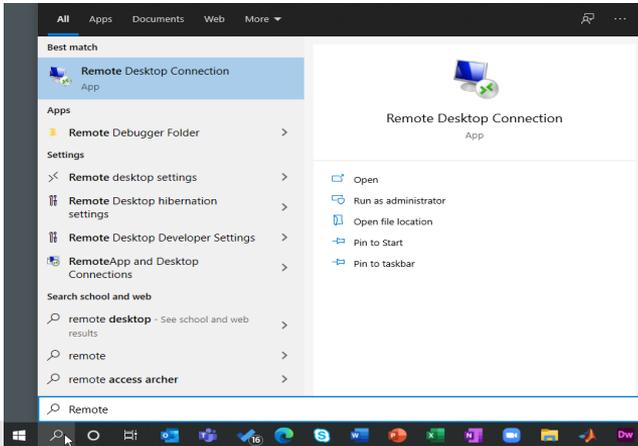
- XRDP is a free and open-source implementation of Microsoft RDP (Remote Desktop Protocol) server. Install it by enter the following:
- `sudo apt-get install xrdp`

2. Open Remote Desktop Connection (RDC) on your Windows Computer. RDS is also available for macOS

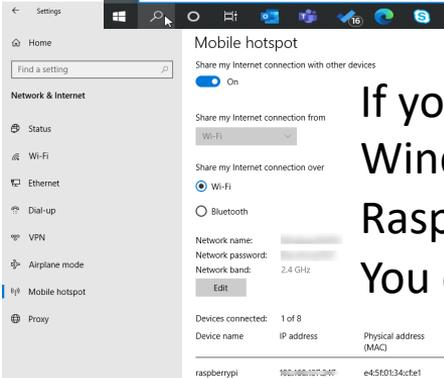
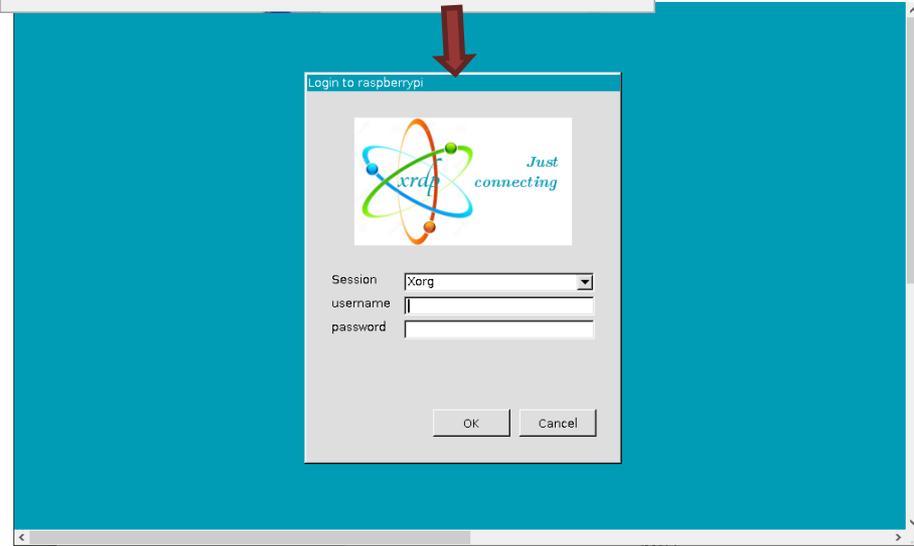
- Enter Computer Name or IP Address
- Default UserName is “pi” and default Password is “raspberrypi” (unless you have changed it)

Alt1: Remote Desktop

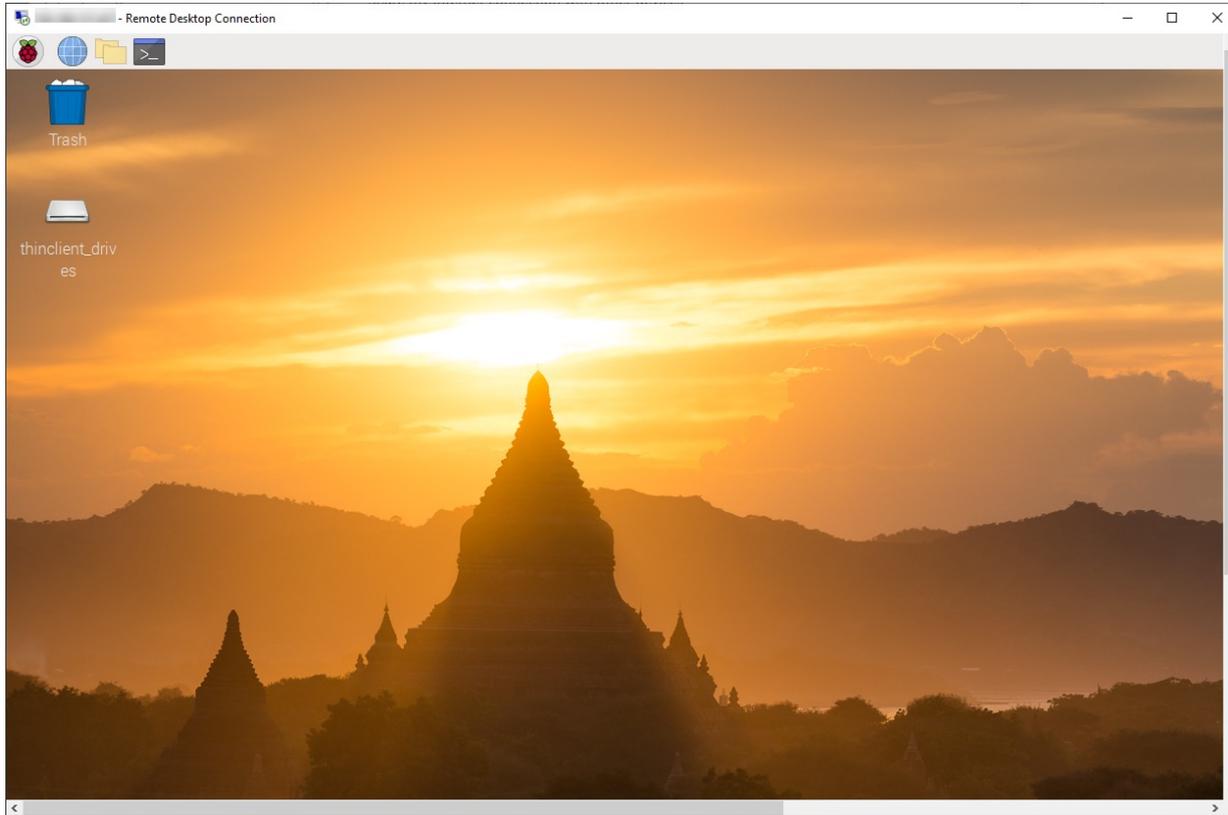
Open Remote Desktop Connection (RDC) on your Windows Computer:



If you use a Mobile hotspot in Windows 10, you can see if your Raspberry Pi is up and running. You can also see the IP address



Alt1: Remote Desktop



You are ready to start using Raspberry Pi remotely from your PC

Alt1: Remote Desktop !

Do you get a **blue screen** after login using RDC/RDP on your Window Computer?

You may need to create another user on your Raspberry Pi. The Pi root user cannot RDP for whatever reason?

- In the Terminal: **sudo adduser <username>**
- Choose password
- Confirm password
- Hit enter for defaults
- Try RDC/RDP again with that login

Alt2: TeamViewer

TeamViewer for Raspberry Pi - Secure, Simple and Powerful - Chromium

teamviewer.com

TeamViewer Products Solutions Service Integrations Partner Company Download Buy now Contact Us

DOWNLOAD / RASPBERRY PI

Windows Mac Linux Chrome OS **Raspberry Pi** Android iOS

TeamViewer for Raspberry Pi - Preview Version

The TeamViewer full client preview version for Raspberry Pi is now available for download.

- Establish incoming and outgoing connections between devices
- Real-time remote access and support
- Start using TeamViewer for free immediately after downloading

[Download 32-bit Preview Version](#) [Download 64-bit Preview Version](#)

There exists a separate version of TeamViewer for Raspberry Pi

Do you want to install this file?

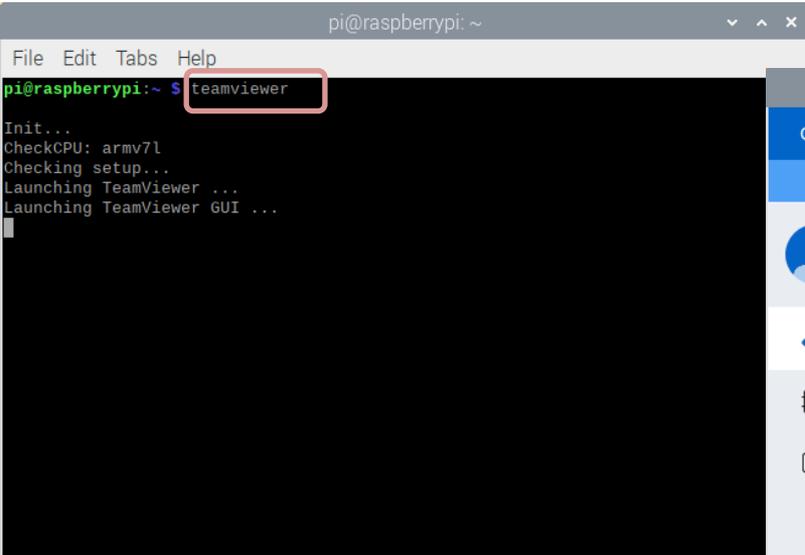
teamviewer_15.22.3_armhf.deb

Cancel Install

<https://pimylifeup.com/raspberry-pi-teamviewer/>

Alt2: TeamViewer

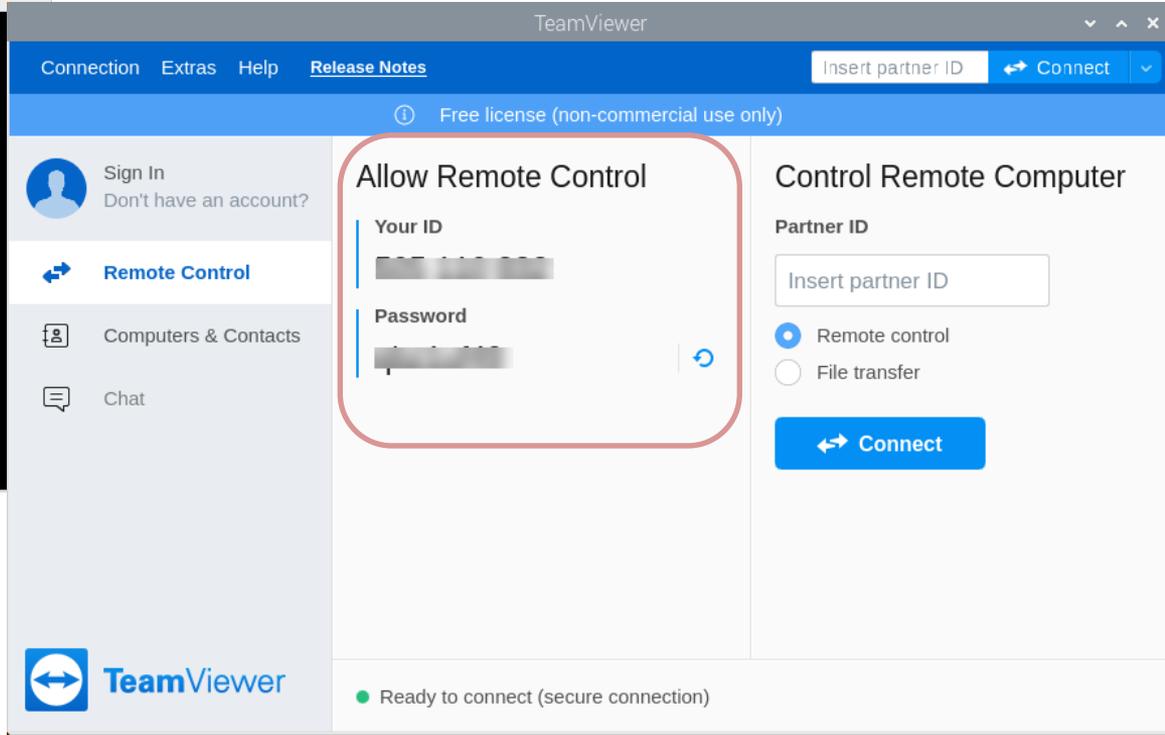
Start TeamViewer on your Raspberry Pi:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ teamviewer  
Init...  
CheckCPU: armv7l  
Checking setup...  
Launching TeamViewer ...  
Launching TeamViewer GUI ...
```

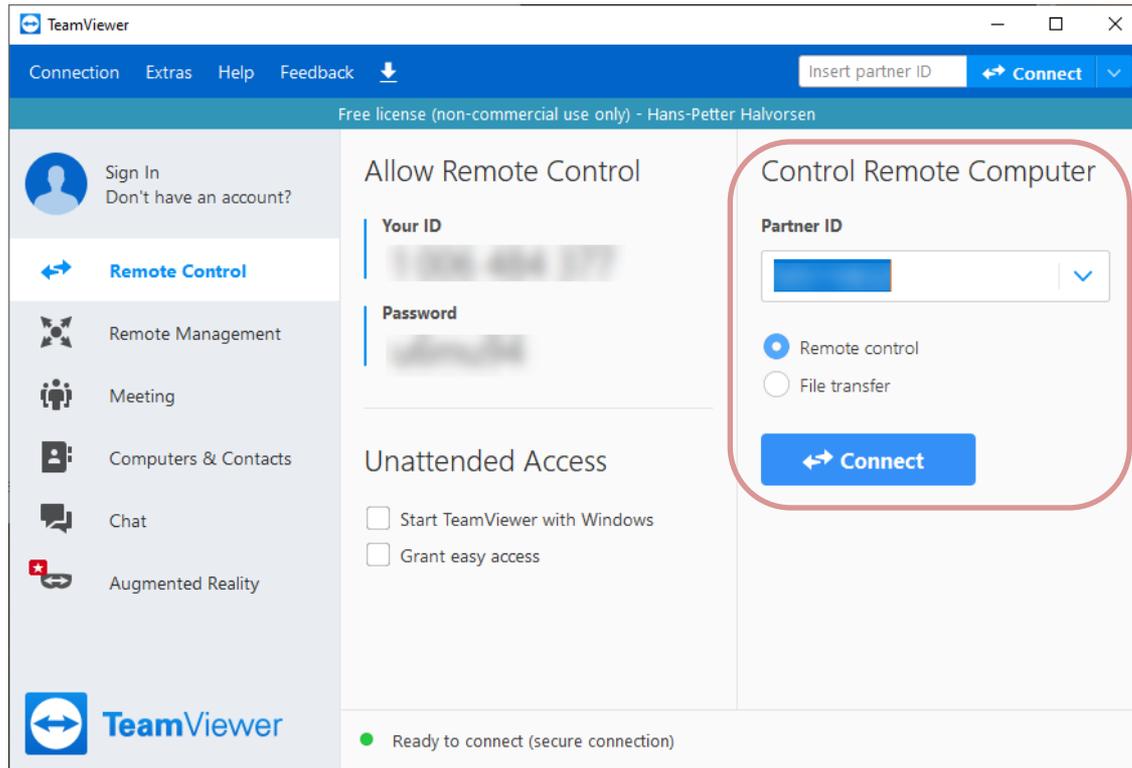
Open TeamViewer by enter
“teamviewer” in the Terminal

Here you see TeamViewer on your Raspberry Pi:



Alt2: TeamViewer

Install TeamViewer on your PC as well. Then use TeamViewer to connect to your Raspberry Pi:





Raspberry Pi and Python Programming

Hans-Petter Halvorsen

[Table of Contents](#)

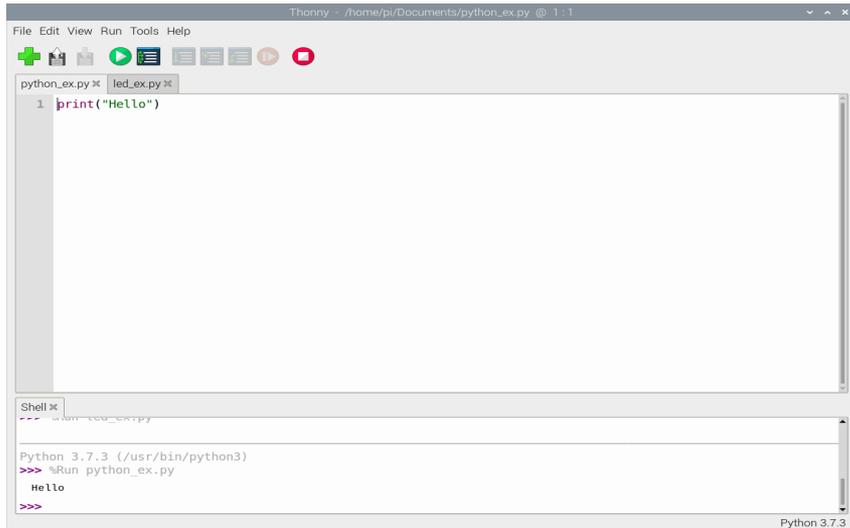
Python with Raspberry Pi

- Python is a fairly old Programming Language (1991) compared to many other Programming Languages like C# (2000), Swift (2014), Java (1995), PHP (1995).
- Python has during the last 10 years become more and more popular.
- Today, Python has become one of the most popular Programming Languages.
- The Raspberry Pi OS comes with a basic Python Editor called “Thonny”

<https://www.raspberrypi.org/documentation/usage/python/>

Python with Raspberry Pi

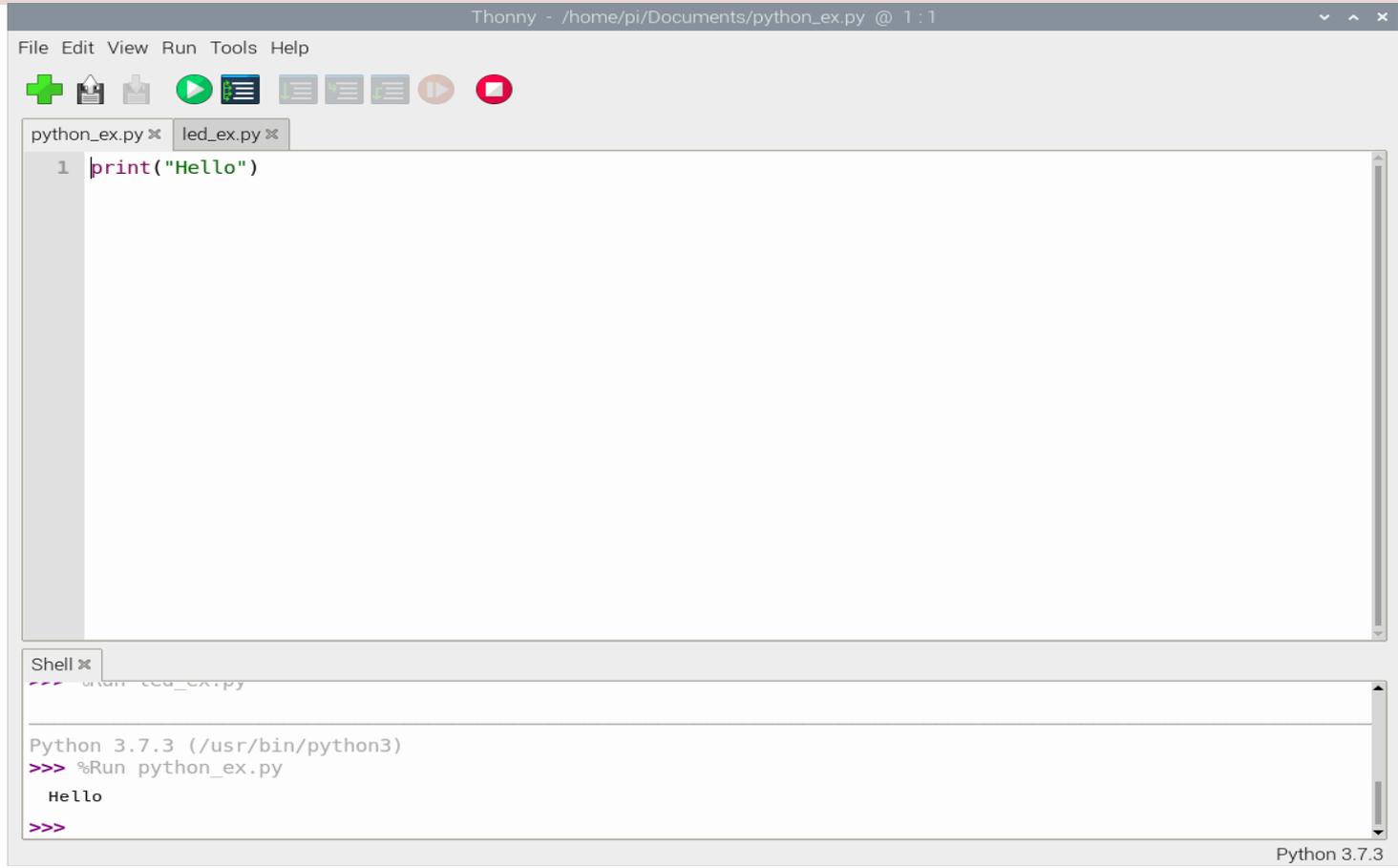
- The Raspberry Pi OS comes with a basic Python Editor called **Thonny**



```
Thonny - /home/pi/Documents/python_ex.py @ 1:1
File Edit View Run Tools Help
python_ex.py x led_ex.py x
1 print("Hello")
Shell x
Python 3.7.3 (/usr/bin/python3)
>>> %Run python_ex.py
Hello
>>>
```

But you can install and use other Python Editors if you prefer

Hello World



The image shows a screenshot of the Thonny Python IDE. The window title is "Thonny - /home/pi/Documents/python_ex.py @ 1 : 1". The menu bar includes "File", "Edit", "View", "Run", "Tools", and "Help". The toolbar contains icons for file operations and execution. Two tabs are open: "python_ex.py" and "led_ex.py". The main editor area shows a single line of Python code: `1 print("Hello")`. Below the editor is a shell window titled "Shell" with the following output: `Python 3.7.3 (/usr/bin/python3)`, `>>> %Run python_ex.py`, `Hello`, and `>>>`. The Python version "Python 3.7.3" is also displayed in the bottom right corner of the IDE window.

Python Packages with Thonny

Tools -> Manage packages...

The screenshot shows the Thonny IDE interface. The 'Tools' menu is open, with 'Manage packages...' selected. Below the menu, a code editor shows a single line of Python code: `print("He`. At the bottom, a shell window displays a `SyntaxError: invalid syntax` and a `NameError: name 'clear' is not defined` after running a script.

The screenshot shows the 'Manage packages for /usr/bin/python3' window. A search bar at the top contains the text 'numpy'. Below the search bar, a list of packages is displayed, with 'numpy' highlighted. To the right of the list, detailed information for 'numpy' is shown, including the installed version (1.16.2), the latest stable version (1.19.4), a summary, author information, and links to the homepage and PyPI page.

numpy

Search on PyPI

lazy-object-proxy
logilab-common
lxml
markupsafe
mccabe
microdotphat
mote
motephat
mypy
mypy-extensions
numpy
oauthlib
olefile
pantilthat
parso
pgzero
phatbeat
pianohat
picamera
piglow
piggpio

Installed version: 1.16.2
Installed to: </usr/lib/python3/dist-packages>

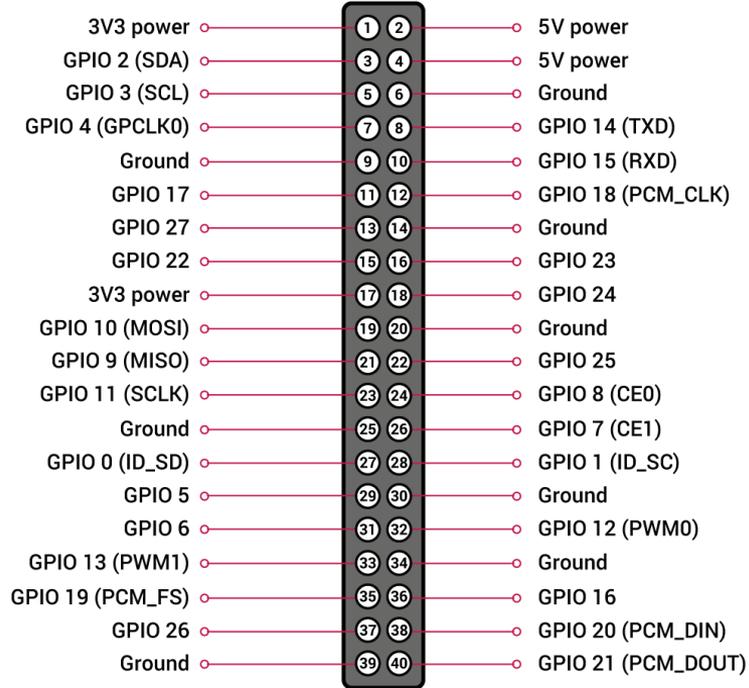
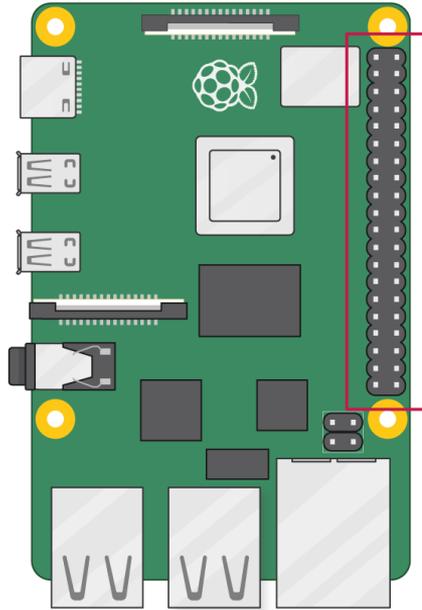
Latest stable version: 1.19.4
Summary: NumPy is the fundamental package for array computing with Python.
Author: Travis E. Oliphant et al.
Homepage: <https://www.numpy.org>
PyPI page: <https://pypi.org/project/numpy/>

Close



GPIO

GPIO



Raspberry Pi is as well suited for prototyping, datalogging and different electronics projects using the GPIO Pins

GPIO Features

The GPIO pins are Digital Pins which are either True (+3.3V) or False (0V). These can be used to turn on/off LEDs, etc.

In addition, some of the pins also offer some other Features:

- PWM (Pulse Width Modulation)

Digital Buses (for reading data from Sensors, etc.):

- SPI
- I2C



GPIO with Python

GPIO with Python

- You can make all kinds of Python program on your Raspberry Pi
- But you could have used your ordinary desktop/laptop PC for that
- The UNIQUE thing with Raspberry Pi compared to an ordinary PC is the GPIO connector
- With GPIO you can connect LEDs, Sensors, control Motors, etc.
- You typically use Python in order communicate with GPIO connector
- That what's makes the combination Raspberry Pi + Python UNIQUE!

GPIO Zero

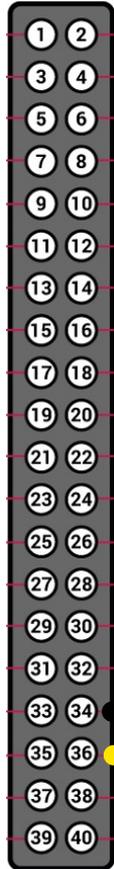
- The GPIO Zero Python Library can be used to communicate with GPIO Pins
- The GPIO Zero Python Library comes preinstalled with the Raspberry Pi OS

Resources:

- <https://www.raspberrypi.org/documentation/usage/gpio/python/>
- <https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/>
- <https://pypi.org/project/gpiozero/>
- <https://gpiozero.readthedocs.io/en/stable/>

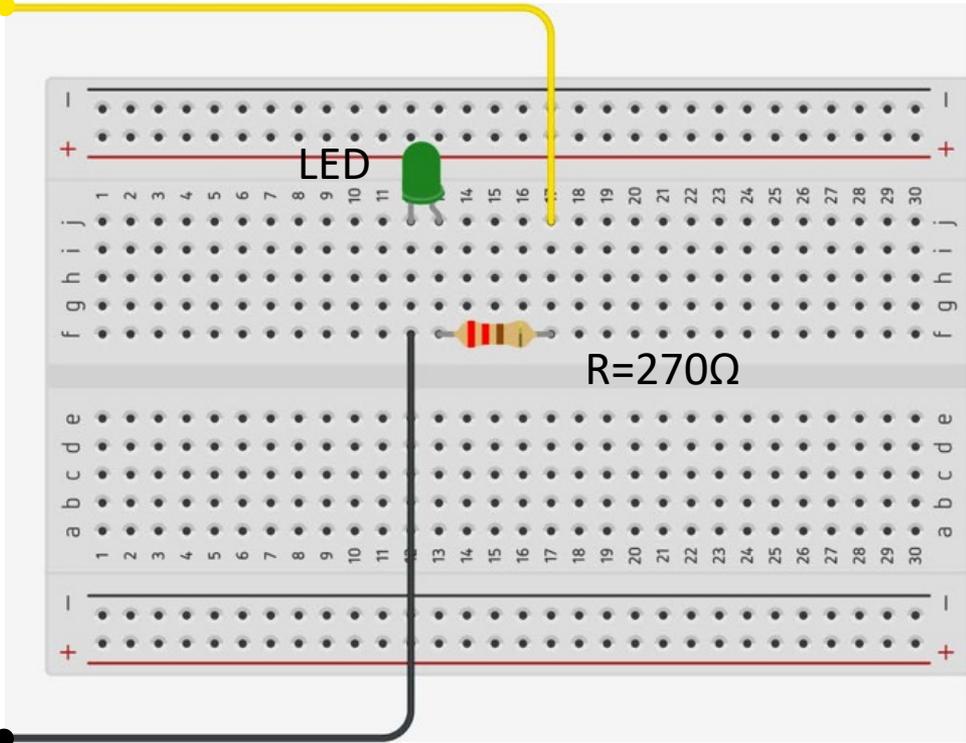
LED Example

Raspberry Pi GPIO Pins



GND (Pin 32)

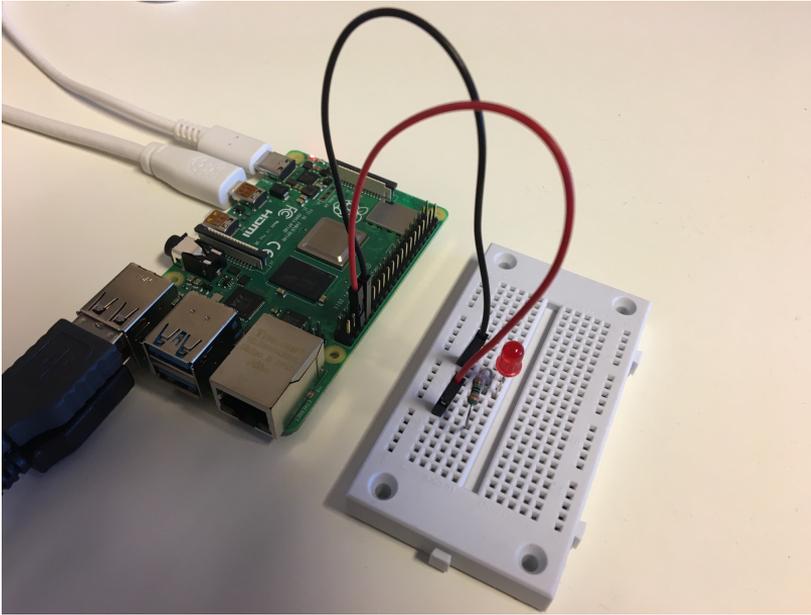
GPIO16 (Pin 36)



Breadboard

LED Example

This Example “Runs for ever”



```
from gpiozero import LED
from time import sleep
```

```
pin = 16
```

```
led = LED(pin)
```

```
while True:
```

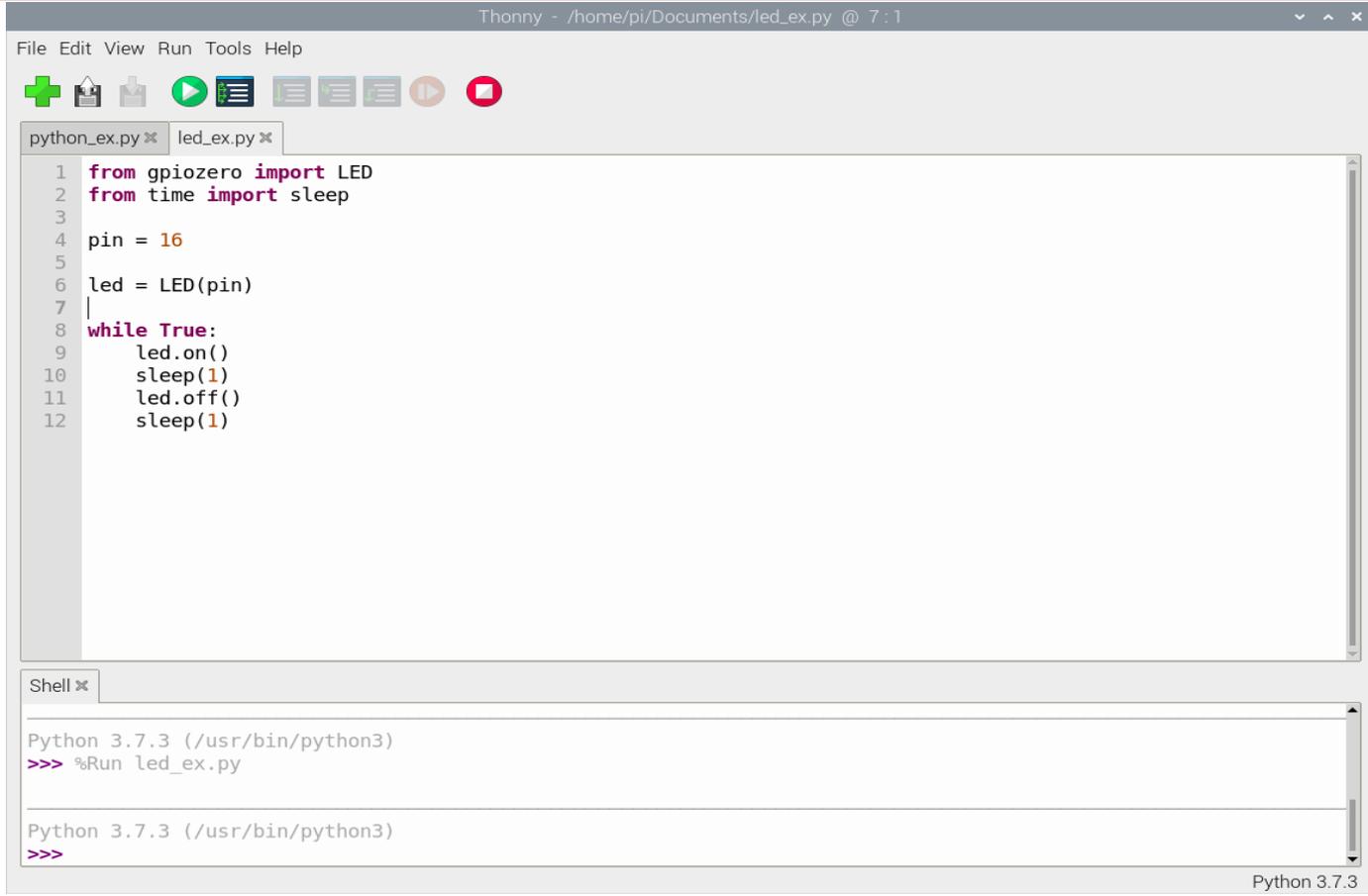
```
    led.on()
```

```
    sleep(1)
```

```
    led.off()
```

```
    sleep(1)
```

LED Example using Thonny



The image shows a screenshot of the Thonny IDE interface. The window title is "Thonny - /home/pi/Documents/led_ex.py @ 7 : 1". The menu bar includes "File", "Edit", "View", "Run", "Tools", and "Help". Below the menu bar is a toolbar with icons for file operations and execution. The main editor area shows a Python script named "led_ex.py" with the following code:

```
1 from gpiozero import LED
2 from time import sleep
3
4 pin = 16
5
6 led = LED(pin)
7
8 while True:
9     led.on()
10    sleep(1)
11    led.off()
12    sleep(1)
```

Below the editor is a shell window with the following output:

```
Shell x
Python 3.7.3 (/usr/bin/python3)
>>> %Run led_ex.py

Python 3.7.3 (/usr/bin/python3)
>>>
```

The Python version "Python 3.7.3" is also displayed in the bottom right corner of the IDE window.



GPIO Features

GPIO Features

In addition to ordinary Digital Pins the Raspberry Pi GPIO connector has other Features as well:

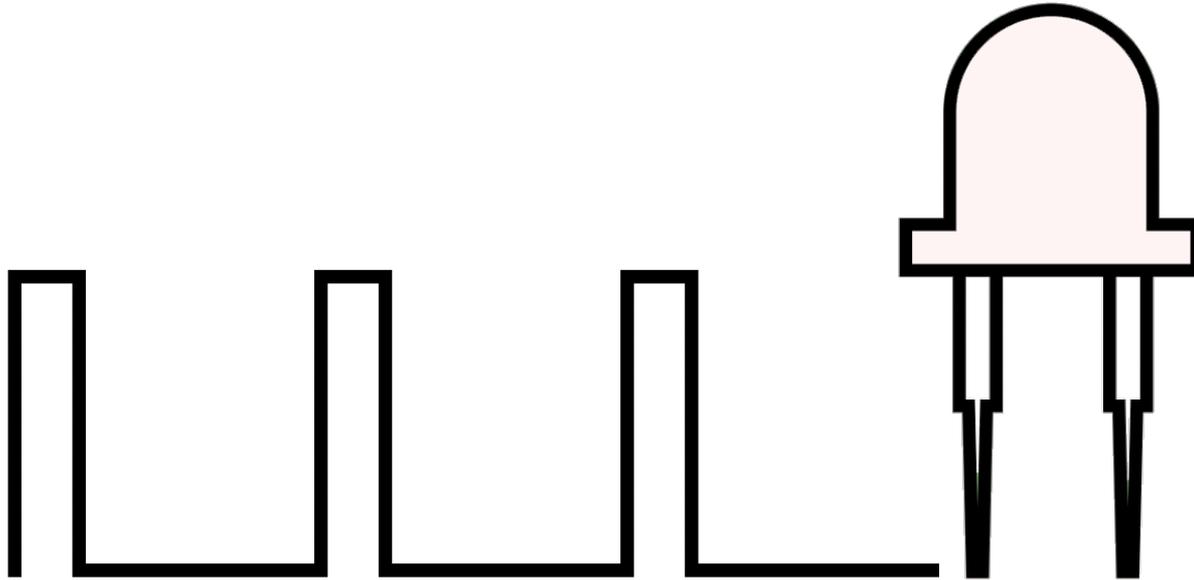
- PWM – Pulse Width Modulation
 - SPI – Serial Peripheral Interface
 - I2C – Inter-Integrated Circuit
- } Communication Interfaces for connection of different types of Sensors, etc.

PWM

- We've seen how to turn an LED on and off, but how do we control its brightness levels?
- An LED's brightness is determined by controlling the amount of current flowing through it, but that requires a lot more hardware components.
- A simple trick we can do is to flash the LED faster than the eye can see!
- By controlling the amount of time the LED is on versus off, we can change its perceived brightness.
- This is known as *Pulse Width Modulation* (PWM).

PWM

Below we see how we can use PWM to control the brightness of a LED



SPI

- SPI is an interface to communicate with different types of electronic components like Sensors, Analog to Digital Converts (ADC), etc. that supports the SPI interface
- Thousands of different Components and Sensors supports the SPI interface

I2C

- I2C is another interface to communicate with different types of electronic components like Sensors, Analog to Digital Converts (ADC), etc. that supports the SPI interface
- Thousands of different Components and Sensors supports the I2C interface



Raspberry Pi with MATLAB

Hans-Petter Halvorsen

[Table of Contents](#)

MATLAB Support Package for Raspberry Pi

The image shows the MATLAB R2020b interface. The top toolbar includes the 'Add-Ons' button, which is circled in red. Below the toolbar, the 'Add-On Explorer' window is open, displaying search results for 'Raspberry Pi'. The search bar at the top of the Add-On Explorer is also circled in red and contains the text 'Raspberry Pi'. The search results are listed below, with the first result, 'MATLAB Support Package for Raspberry Pi Hardware', highlighted by a red box. This result includes a star rating, download count (285), and update date (14 Oct 2020). The second result, 'Simulink Support Package for Raspberry Pi Hardware', also includes a star rating, download count (185), and update date (14 Oct 2020). The third result, 'Raspberry Pi Hardware Resource Manager', includes a star rating, download count (89), and update date (29 Jul 2019).

Current Folder: C:\Users\hansha\Documents\MATLAB

Command Window: New to MATLAB? See resources for [Getting Started.](#)

Add-On Explorer Search: Raspberry Pi

MATLAB Support Package for Raspberry Pi Hardware by MathWorks
MATLAB Hardware Team **STAFF**
285 Downloads ⁱ
Updated 14 Oct 2020
Acquire sensor and image data from your Raspberry Pi.
MATLAB® Support Package for Raspberry Pi™ Hardware enables you to communicate with a Raspberry Pi remotely from a computer running MATLAB. You can acquire data from sensors and imaging devices.
Hardware Support

Simulink Support Package for Raspberry Pi Hardware by MathWorks
Simulink Team **STAFF**
185 Downloads ⁱ
Updated 14 Oct 2020
Run models on Raspberry Pi.
Simulink® Support Package for Raspberry Pi™ Hardware enables you to create and run Simulink models on Raspberry Pi hardware. The support package includes a library of Simulink blocks for configuring
Hardware Support

Raspberry Pi Hardware Resource Manager version 1.0 by MathWorks
Simulink Team **STAFF**
89 Downloads ⁱ
Updated 29 Jul 2019
Monitor the status of different hardware resources on the raspberry pi

Getting Started with MATLAB Support Package for Raspberry Pi: <https://youtu.be/32ByiUdOsw>

Test Hardware

The screenshot shows the MATLAB R2020b interface with the Command Window displaying the output of the `raspi` command. The Command Window shows the following output:

```
>> r = raspi

r =

raspi with properties:

    DeviceAddress: '172.20.10.11'
        Port: 18734
    BoardName: 'Raspberry Pi 4 Model B'
    AvailableLEDs: {'led0'}
AvailableDigitalPins: [4, 5, 6, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]
AvailableSPICChannels: {'CE0', 'CE1'}
    AvailableI2CBuses: {'i2c-1'}
    AvailableWebcams: {}
        I2CBusSpeed: 100000

Supported peripherals
```

The Workspace window shows a variable `r` of type `1x1 raspi`.

Name	Value
r	1x1 raspi

Documentation and Examples

Documentation

CONTENTS

« Documentation Home

Category

- Installation and Setup
- Connection to Raspberry Pi Hardware
- Run on Target Hardware
- LEDs
- GPIO Pins
- Serial Port
- I2C Interface
- SPI Interface
- Camera Board
- Sense HAT
- Web Camera
- Pulse Width Modulation
- Servo
- Linux
- Display
- Audio

All Examples Functions

MATLAB Support Package for Raspberry Pi Hardware

Program sensor and image applications on Raspberry Pi

MATLAB® Support Package for Raspberry Pi™ Hardware enables you to communicate with a Raspberry Pi remotely from a computer running MATLAB or through a web browser with MATLAB Online™. You can acquire data from sensors and imaging devices connected to the Raspberry Pi and process them in MATLAB. You can also communicate with other hardware through the GPIO, serial, I2C, and SPI pins.

The support package functionality is extended if you have MATLAB Coder™. With MATLAB Coder, you can take the same MATLAB code used to interactively control the Raspberry Pi from your computer and deploy it directly to the Raspberry Pi to run as a standalone executable.

 [Release Notes](#)
 [PDF Documentation](#)

Installation and Setup

Install support for the hardware, update the firmware, and connect to the hardware

Connection to Raspberry Pi Hardware

Create a connection to Raspberry Pi hardware

Run on Target Hardware

Deploy a MATLAB function as a standalone executable on the hardware

LEDs

Use the Raspberry Pi's LED

GPIO Pins

Use the Raspberry Pi's GPIO pins

Serial Port

Use the Raspberry Pi's serial port

I2C Interface

Use the Raspberry Pi's I2C interface

Blinking LED Example

The image shows the MATLAB R2020b interface. The main editor window displays a script named `blinkLED.m` with the following code:

```
1 function blinkLED()  
2  
3     r = raspi;  
4  
5     for i = 1:10  
6         disp(i);  
7         writeLED(r, "LEDO", 0);  
8         pause(0.5);  
9         writeLED(r, "LEDO", 1);  
10        pause(0.5);  
11    end  
12  
13 end  
14
```

The Command Window at the bottom shows the execution of the script:

```
>> edit blinkLED.m  
>> blinkLED  
1  
2  
3  
4
```

A red callout box on the right side of the interface contains the text: "Observe that the built-in LED on the Raspberry Pi is blinking".

Blinking LED Example

We use the following Function:

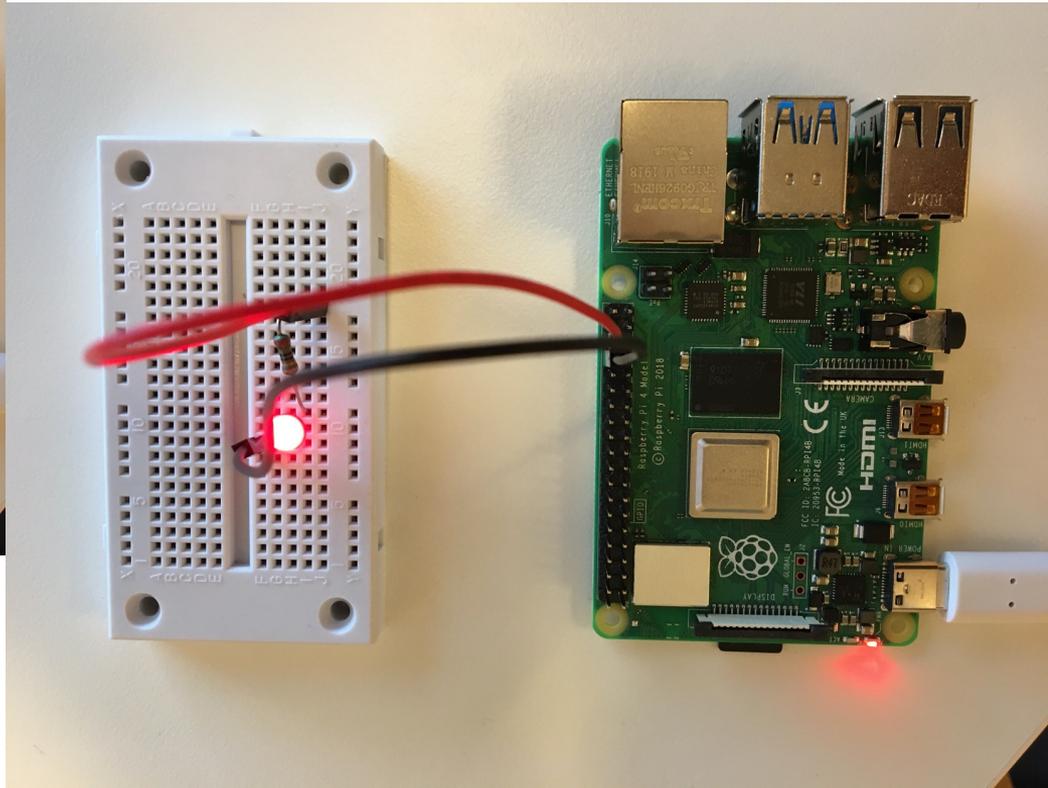
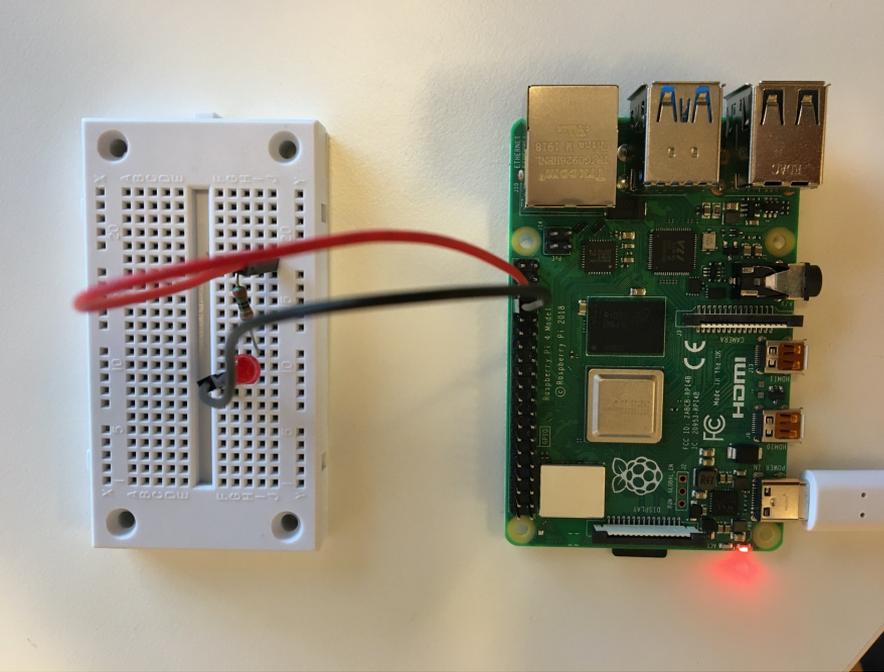
```
writeLED(r, "LED0", 1);
```

```
clear
clc

r = raspi;

for i = 1:10
    disp(i);
    writeLED(r, "LED0", 0);
    pause(0.5);
    writeLED(r, "LED0", 1)
    pause(0.5);
end
```

Blinking LED GPIO Example



Blinking LED GPIO Example

```
clear rpi
rpi = raspi();
gpiopin = 16
ledvalue = 1;
writeDigitalPin(rpi, gpiopin, ledvalue);

for i = 1:10
    ledvalue = 1;
    writeDigitalPin(rpi, gpiopin, ledvalue);
    pause(0.5);
    ledvalue = 0;
    writeDigitalPin(rpi, gpiopin, ledvalue);
    pause(0.5);
end
```



ThingSpeak

With Raspberry Pi and Python

Hans-Petter Halvorsen

[Table of Contents](#)

ThingSpeak

- ThingSpeak is a IoT Cloud Service that lets you collect and store sensor data in the cloud and develop Internet of Things applications.

<https://thingspeak.com>

- It works with Arduino, Raspberry Pi and MATLAB, etc.
- ThingSpeak Raspberry Pi Python Library:

<https://pypi.org/project/thingspeak/>

Write TMP36 Data

Python Example
running on Raspberry Pi

```
import thingspeak
import time
from gpiozero import MCP3002

adc = MCP3002(channel=0, differential=False)

channel_id = xxxxxxxx
write_key = "xxxxxxxxxxxxxxxxxxxxxx"

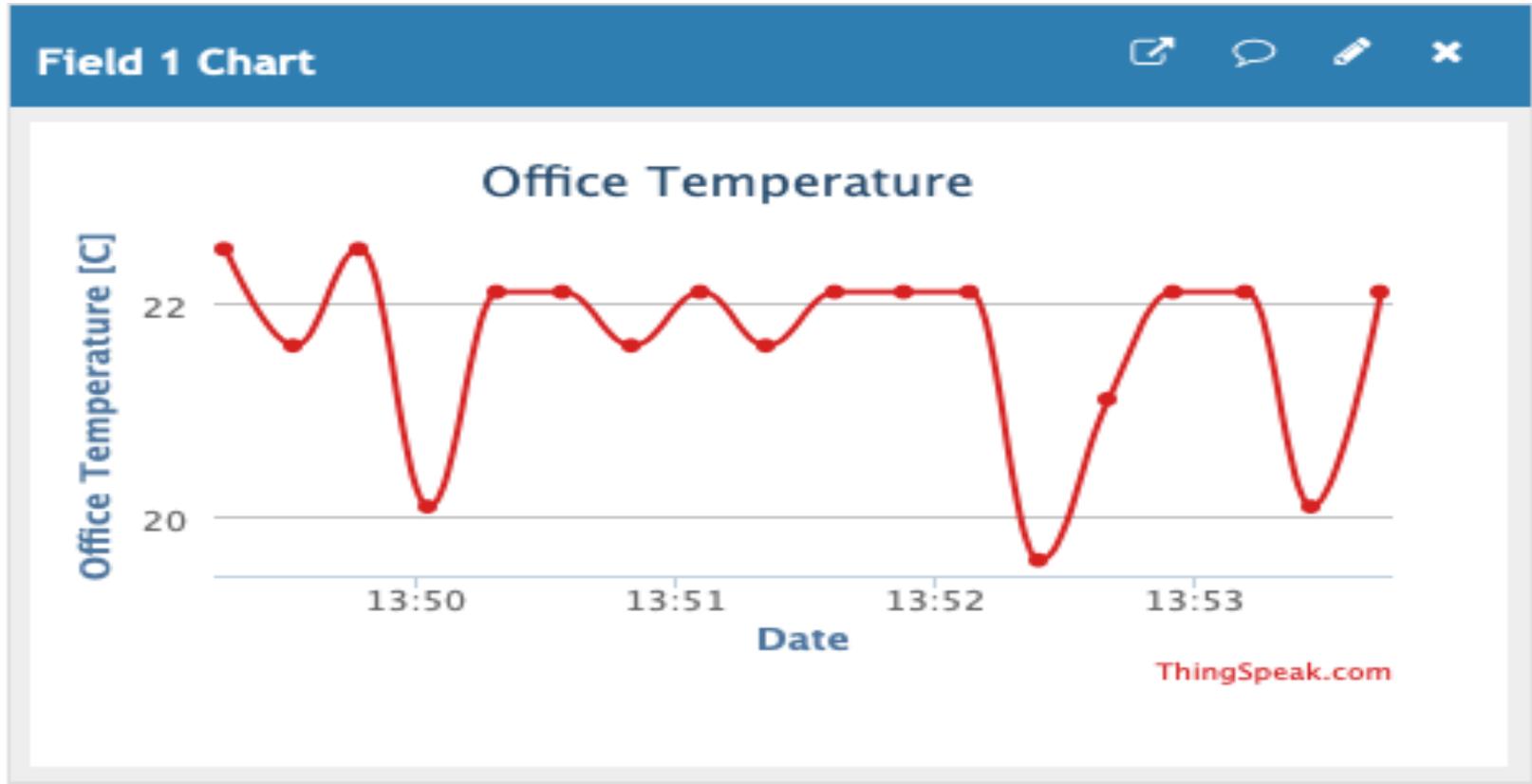
channel = thingspeak.Channel(id=channel_id, api_key=write_key)

N = 10
for x in range(N):
    #Get Sensor Data
    adcddata = adc.value #Scaled Value between 0 and 1
    voltvalue = adcddata * 5 # Value between 0V and 5V
    tempC = 100*voltvalue-50 # Temperature in Celsius
    tempC = round(tempC,1)
    print(tempC)

    #Write to ThingSpeak
    response = channel.update({'field1': tempC})
    time.sleep(15)
```

Write TMP36 Data

Here we see the Temperature Data in ThingSpeak:



Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

