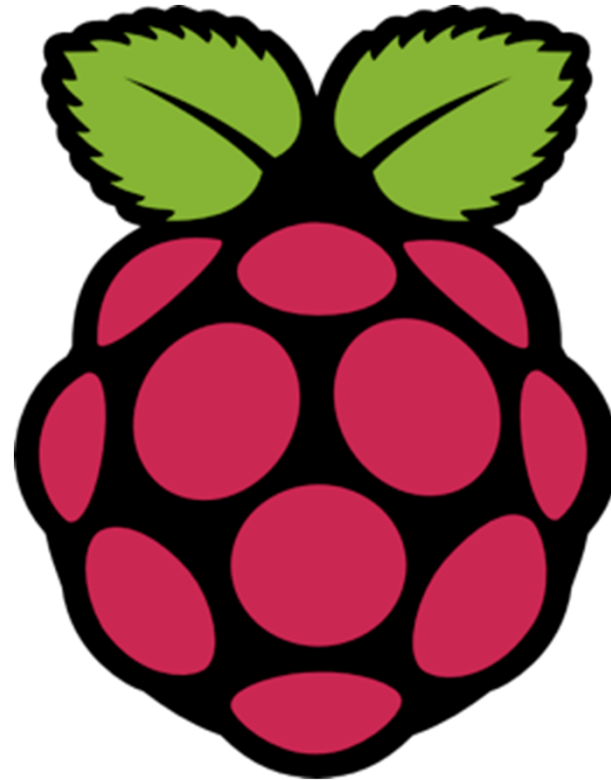# Raspberry Pi and Samba

Francesco Zampognaro

Marco Bonola
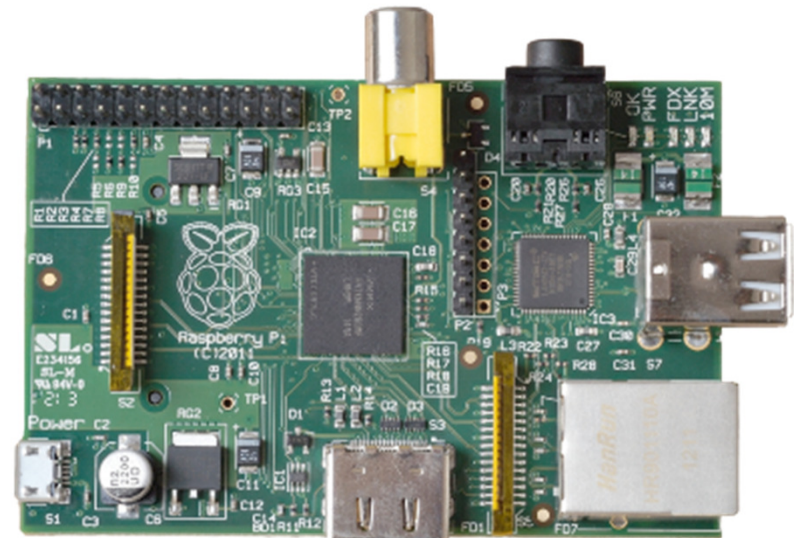
RaspberryPi

In March 2019, sales reached 25 million units sold

What is a Raspberry Pi?

- Raspberry Pi Foundation (UK, 2008): first board introduced in 2012 to promote teaching and education activities
- Full fledge "PC" in credit card size
- Plugs into a TV or monitor
- Cheap and available
- Capability:
  – Programming
  – Electronic Projects
  – Office
  – Play HD Videos

Versions of Raspberry Pi

- Hardware platform
  - Raspberry Pi Zero (low cost low specs)
  - Raspberry Pi
  - Raspberry Pi 2
  - Raspberry Pi 3 (with Wifi + Bluetooth)

 Model A (no Ethernet) or B

- Operating System
  - Raspbian (officially supported), based on Debian
    - Noobs installer
  - 3rd party OS
    - https://www.raspberrypi.org/downloads/ (Win10 IoT core, Ubuntu Mate, etc.)
  - Many others
    - https://en.wikipedia.org/wiki/Raspberry_Pi#Software

Programming Languages

- We have a full (almost) operating system!
- Any language which will compile for ARM can be used
- Installed by default on the Raspberry Pi Raspbian, including drivers and libraries for I/O:
  - C
  - C++
  - Java
  - Scratch
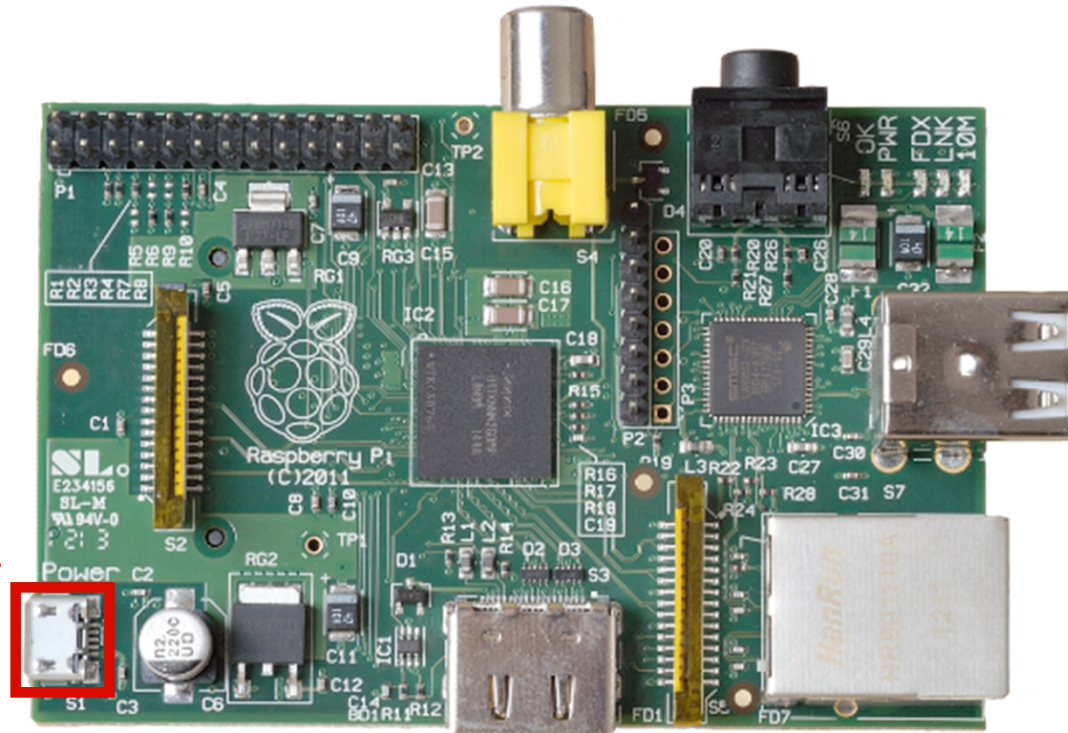 - Ruby
  - **Python**

What we have here?

- Raspberry Pi model B/B+ Discontinued

- To buy now:
  - Raspberry Pi3 Model B+ (35€) Ethernet
  - Raspberry Pi3 model A (40€) wifi
  - Raspberry Pi Zero Wifi (10€, kit 25€)
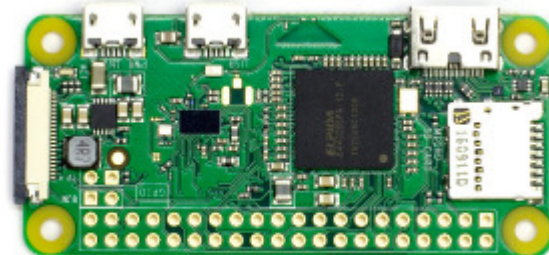
Power
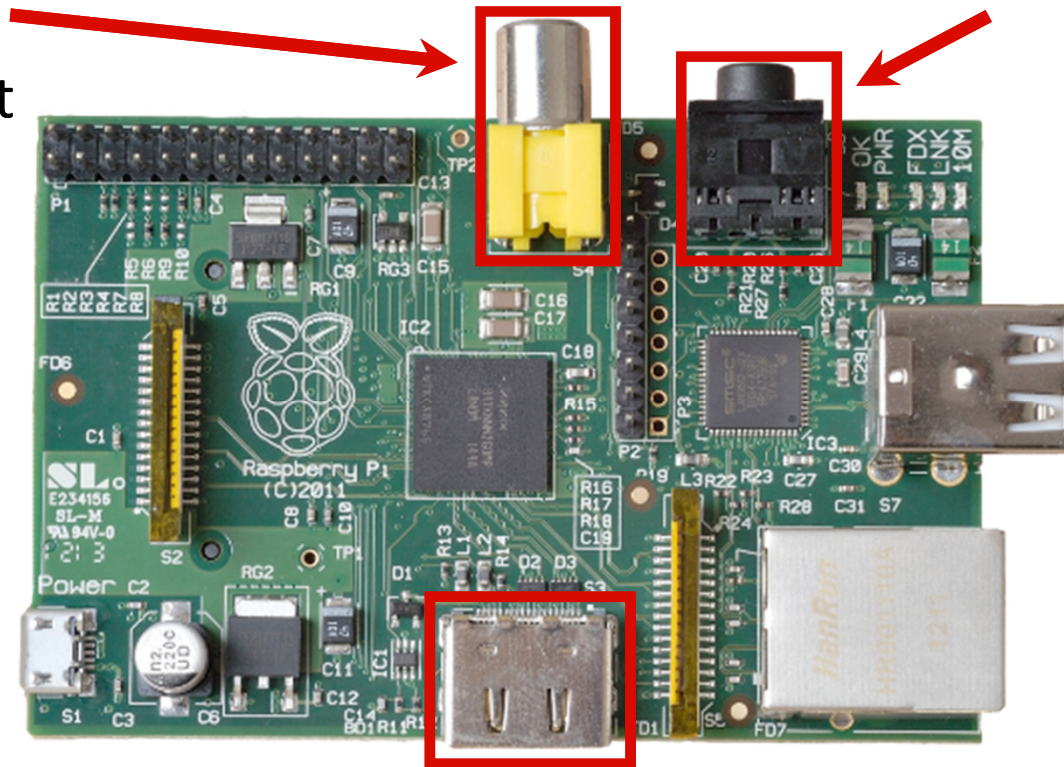

5v micro
USB
connector

Layout and I/O slightly different based on model!

A/V (Audio/Video)

3.5mm Audio Standard headphone socket

RCA Video (works with most older TVs)

HDMI Audio & Video (Full HD digital output!)

Connectivity

2 x USB 2.0 ports

GPIO (General Purpose Input & Output)
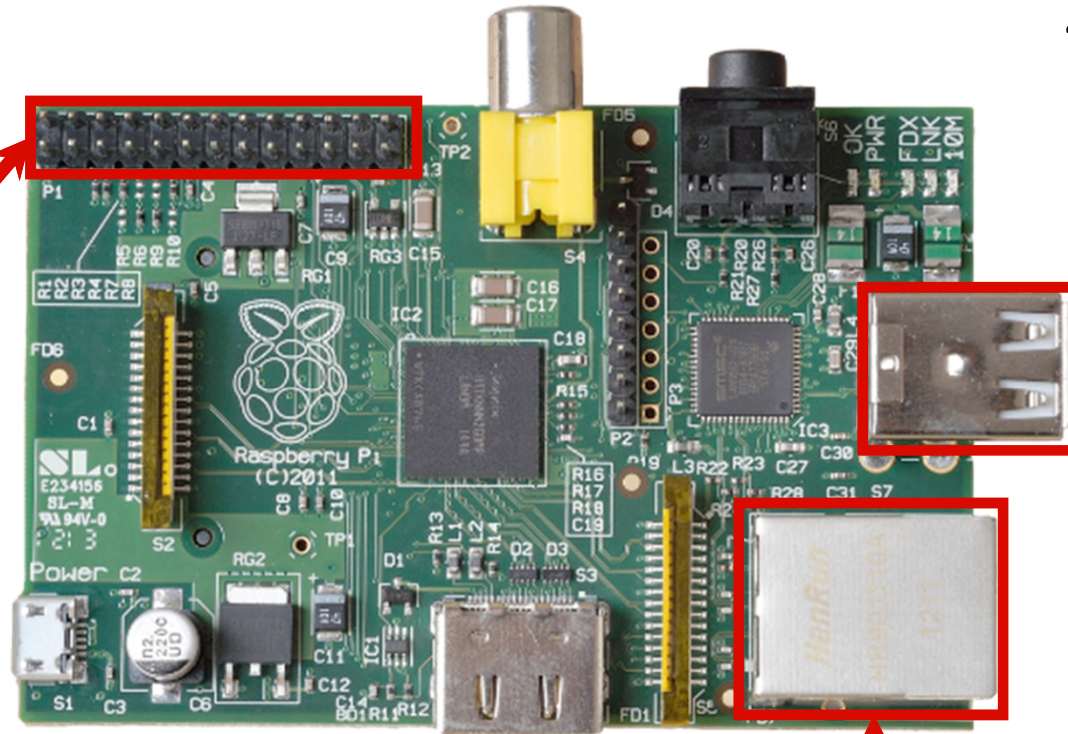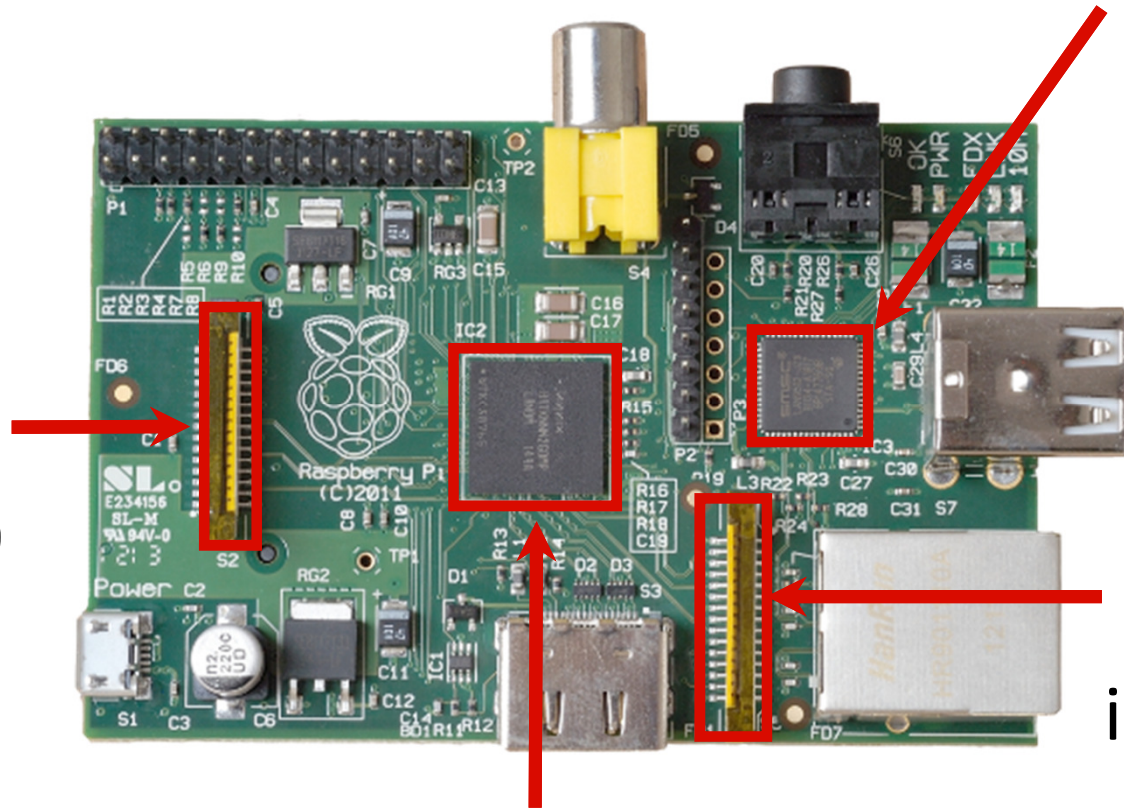
10-100 Mbit/s Ethernet (on model B)

Other I/O and chips

LAN Controller

DSI
(display
interface)

CSI
(camera
interface)

SOC (System On a Chip)
Broadcom BCM2835 700Mhz & RAM

On the back: Storage
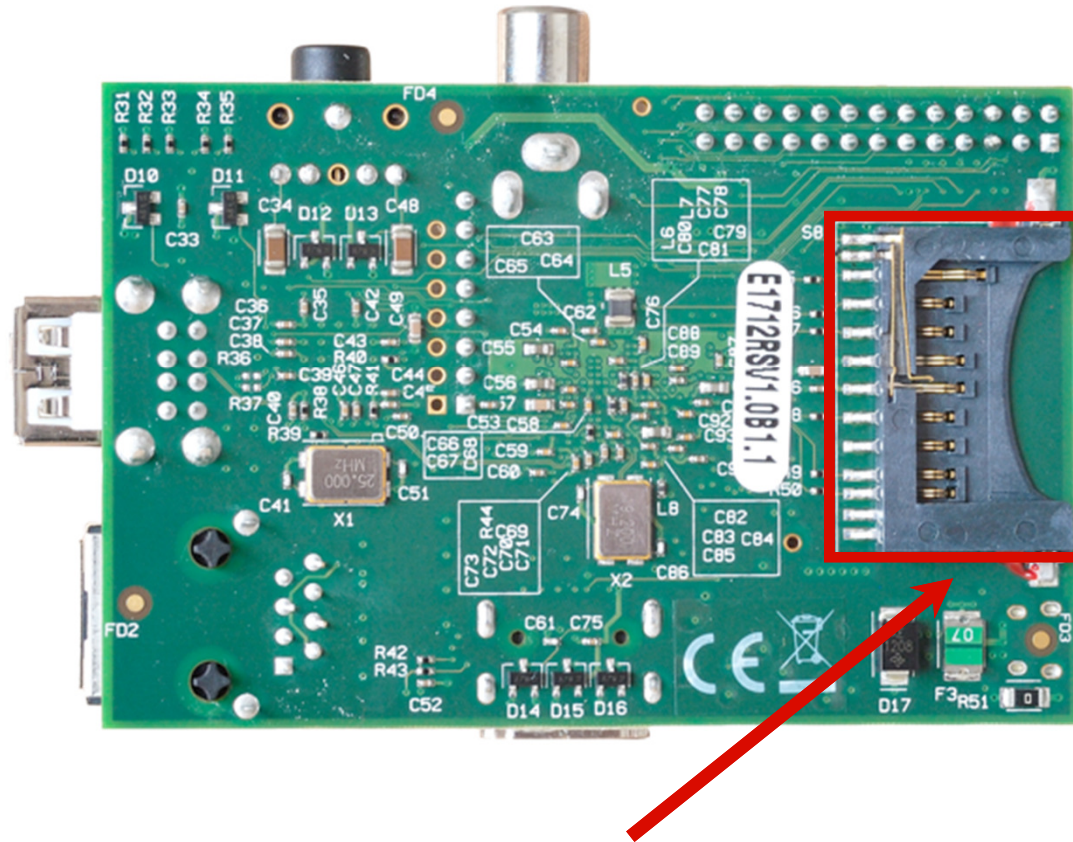
SD Card Slot
(supports SD cards up to 32GB)

Installing O.S.

- OS must run on a dedicated (micro-) SD card. If we don't have a pre-installed SD, we must create one.
  - Download Raspbian ISO, or any other compatible OS
  - Use a software to create a bootable SD card with that iso:
    - Etcher (windows)
    - Commandline dd (linux)
- For Raspbian it is possible to simply format SD and move Noobs files inside:
  - https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/3

Strenghts?

- Many other low-cost single-chip PC exist (Chromecast, cheaper, Intel NUC more expensive). Why RaspPi?
  - Widespread diffusion, standard (and cheap) HW
  - Standard operating system, standard connectors! Allows to do almost everything for small-scale needs (web server, file server, multimedia station)
  - **GPIO**: general purpose I/O pins:
    - Extensions through 3rd party expansion boards (or "shields")
      - https://www.cooking-hacks.com/shop/raspberry-pi/shields
    - **Includes SPI, I2C, (1-wire) interfaces**: cheap touch screen... and sensors!!
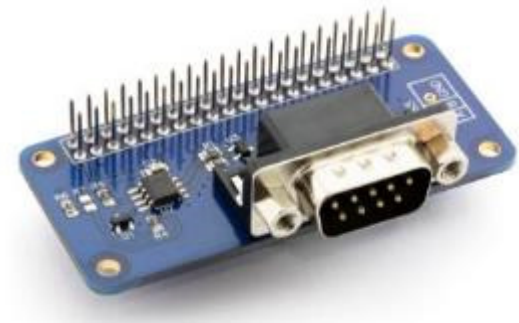
GPIO

Raspberry Pi B
Rev 2 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

Key

Power +	UART
GND	SPI
I²C	GPIO

Raspberry Pi B+
B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

GPIO

- Example: put a 5 V relè driven by GPIO 12. HW setup:
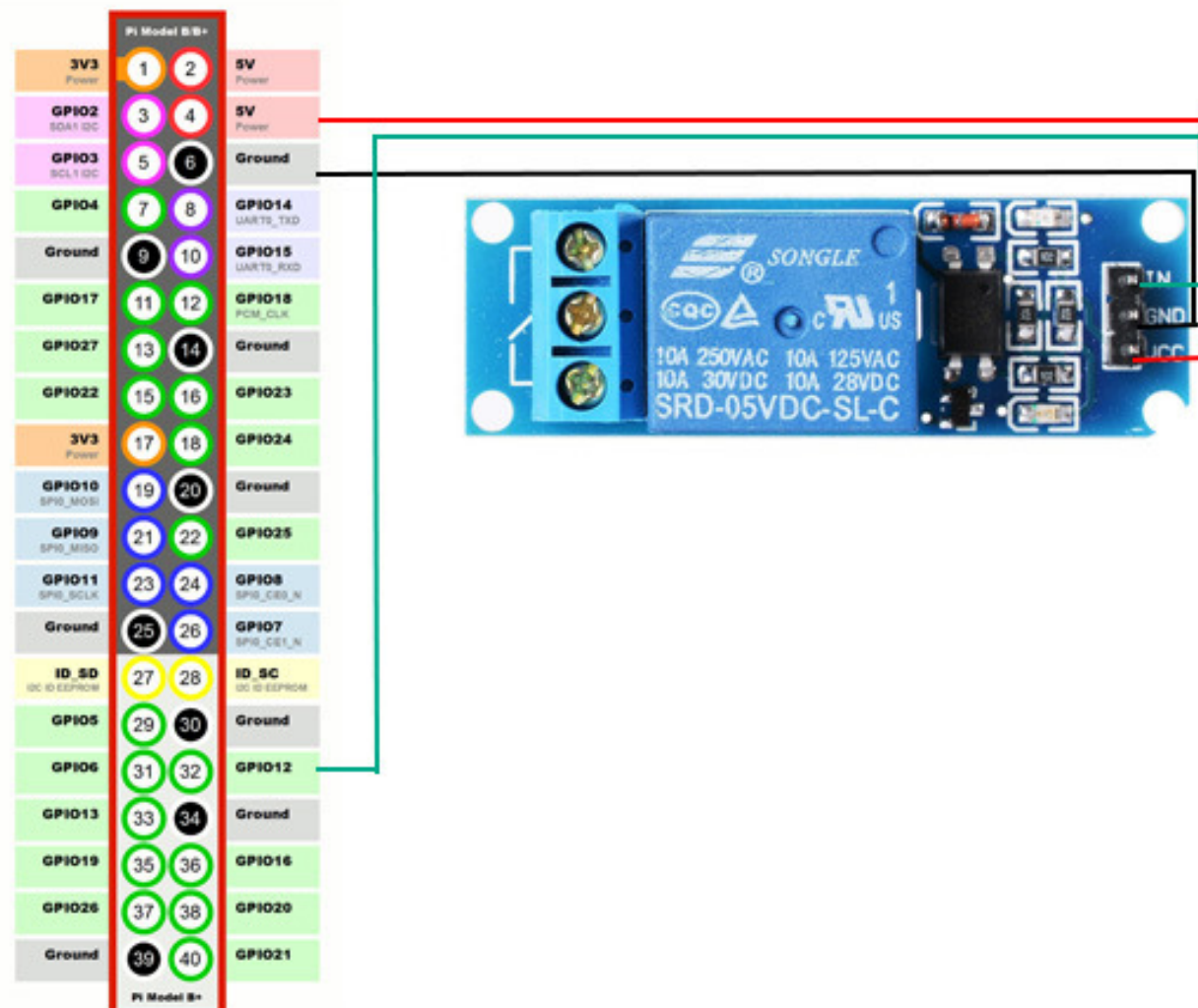
GPIO

- The software.
  - Python:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
GPIO.output(12, GPIO.HIGH)
GPIO.output(12, GPIO.LOW)
```
  - C:

```
#include <wiringPi.h>
…
```
  - BASH:

```
echo 12 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio12/direction
echo 1 > /sys/class/gpio/gpio12/value
echo 0 > /sys/class/gpio/gpio12/value
```

SW

- All linux services seen so far (including next Samba) are available with simple "debian-like" installation
  - File server
  - Print server
  - FTP server
- Dynamic web pages (ie, python) can interact with GPIO to create home automation or custom configurations!
  - Thermostats
  - Remote-activated electrical operations (open a gate, turn on a light, etc.)

What is Samba?

- Upon consolidation of networking, Microsoft introduced **Server Message Block** (SMB) protocol for resource sharing in LAN PCs.

- Integrated as part of a suite to manage Windows domains (WINS) and on top of NetBIOS. When switching to TCP/IP, it used port 139

- As TCP/IP became the de-facto standard, SMB **Common Internet File System** (CIFS) was defined (TCP port 445). Then SMB1/CFS, SMB2 and today SMB3 (security!)

What is Samba?

❖ We want to make Linux and Windows friends!!

- SMB → Samba
  (https://www.samba.org/samba/docs/)
  - Free suite of programs that enables flavors of UNIX to work with other operating systems such as OS/2 and Windows, as both a server and a client
  - As a server, Samba shares Linux files and printers with Windows systems.
  - As a client, Samba gives Linux users access to files on Windows systems (or Linux systems using Samba).

Shares

- With Linux-Samba we can share a directory hierarchy
- Equivalent native service for linux is called NFS – Network File System (where a share is an "export").
- It is also possible to share printers (Network printers)
- It is possible to allow listing of available shares, and restrict access on specific rules

Install

- apt-get install samba samba-client
- Configuration file in: /etc/samba/smb.conf
- <u>minimal example</u>:

```
[global]
    workgroup =  CGRL
    security = user
    browsable = yes
    local master = yes
[share_name]
    path = /tmp
    read only = yes
    public = yes

[homes]
    guest ok = no
    read only = no
    browseble = no
[printers]
    path = /var/spool/samba/
    printable = yes
    browseble = no
    valid users = francesco cgrl
```

stanza

mkdir -p /var/spool/samba/
# chmod 1777 /var/spool/samba/

Samba Access

- For a user to gain access to Samba services on a Linux system, he/she must provide a username and password.
  - It is also possible to authenticate using other methods such as LDAP or PAM
  - It is possible to define samba-specific users or leverage Linux users; both must have associated Samba passwords
  - Possible anonymous read-only access

Username and password

- The supplied username must be "valid" for non-guest access. Possible restrictions on shares based on username are possible.

- security = user
  - requires a valid Linux account

- But passwords are handled differently. We need to do the command:
  - smbpasswd -a <username>

- Password are store to a local file, specified in smb.conf

Access to the shared res

- When you attempt to connect from Windows to a Samba server, Windows presents your Windows username and password to Samba. If your windows username is the same as the Linux username, and if your Windows and Samba passwords are the same, you do not have to enter a username and password to connect to the Samba server.

- Otherwise a login page appears.
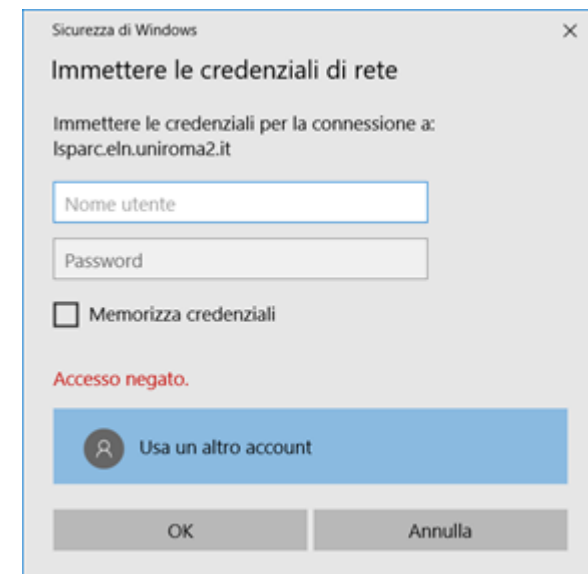
Accessing Windows Shares from Linux

- As a client, Samba enables you to view and work with files on a Windows system (or Linux Samba server) from a Linux system.

- In Linux SMB shares are indicated as

    [smb:]//IP-address/share_name

- `smbtree` – displays windows shares in a hierarchical tree (auto search on the connected networks). When prompted for a password, hit enter to view shares available to guest. Enter password to view restricted shares.

smbclient: Connects to Windows Shares

- *smbclient:* similarly to an ftp client, connects to a SMB share to perform commandline operation (get a file, store a file, get list of directories etc.)

- `$>` `smbclient //winhost/share` connects to the share folder "share" (maybe its home, or a dedicated share). Once the access is granted, we have an interactive prompt :

  - smb:\>

- `$>` `smbclient -L //winhost` shows the list of shares.

Mounting Windows Shares

- On a Linux PC, you can "mount" a Windows share so that you can access it as a local filesystem.

- `mkdir share`

- `mount [-t cifs] //host/sharedir /share -o username=mark,password=pizza,rw`

- `ls /share`

Troubleshooting

- Samba provides some utilities to troubleshoot a connection:
  - `smbstatus` – displays a report on open Samba connections
  - `testparm` – checks the syntax of /etc/samba/smb.conf and displays its contents
- From Windows, to display a list of shares available from the server
  - `net view \\servername`

Samba Lab
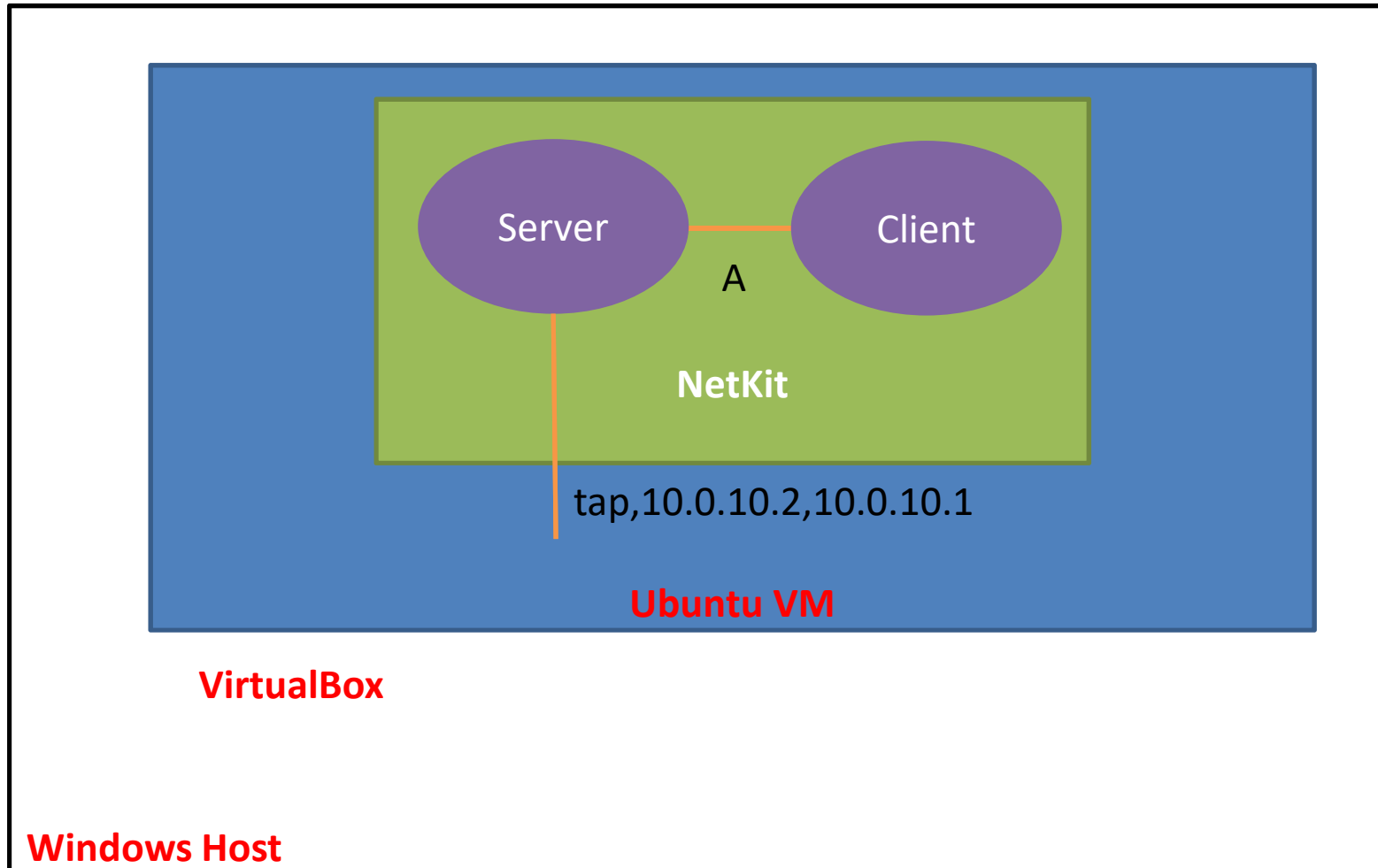
Edit the startup of server

- Create a system user (not the only option, but most common)
- Create a samba password for that user (not interactive)
- Start service (in can be smbd or samba)

```
useradd cgrl
echo -ne "cgrlpass\ncgrlpass\n" | smbpasswd -s -a cgrl
/etc/init.d/samba start
```

- Where is password file: let's check

```
testparm -v | grep passwd
```

  - /var/lib/samba/private/smbpasswd
  - Default: /etc/samba/smbpasswd

- (in case of password backhand = tdbsam file is at /var/lib/samba/passdb.tdb. But we will revert to plain file for simplicity (default))

- **Create folders and files**

```
mkdir -p server/etc/samba
vi smb.conf

[global]
    workgroup =  CGRL
    security = user
    browsable = yes
    local master = yes
[cgrl_share]
    path = /tmp
    read only = no
    public = no
    valid users = cgrl
```

```
lstart
```

- From client check everything is fine

```
smbclient -L //10.0.0.1 -U cgrl
```

- Let's mount it!

```
mount [-t cifs]
//10.0.0.1/cgrl_share  /mnt -o
username=cgrl,password=cgrlpass,rw
ls /mnt
```

- Access from Ubuntu host (through TAP interface).

- warning:
  - files created through the share will have as owner "cgrl" user, defined by us
  - files created directly in netkit server are owned by root