



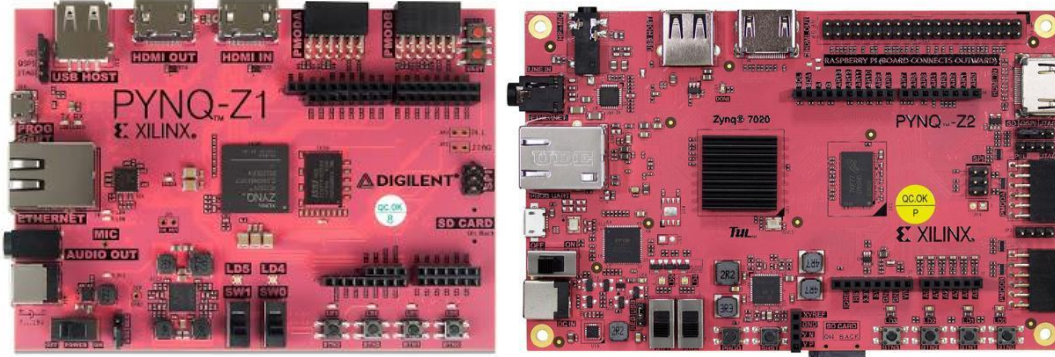
Getting started



Agenda

- > Configure board, SD card
- > Connect to the board
- > Login in to portal
- > Jupyter Notebook

Prerequisites

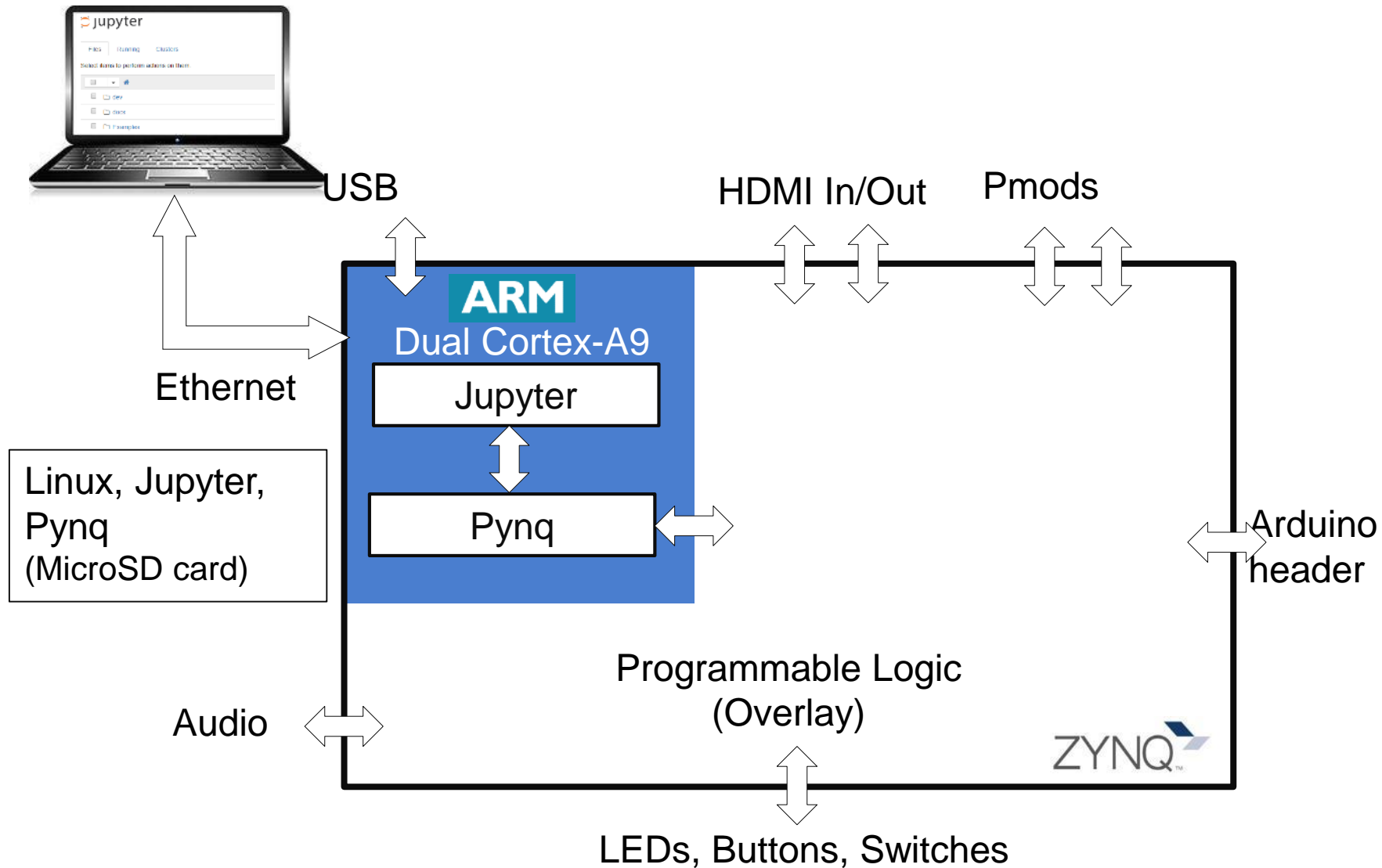


- > PYNQ board
- > SD Card with preloaded Image*
- > Ethernet Cable
- > USB cable and available USB port



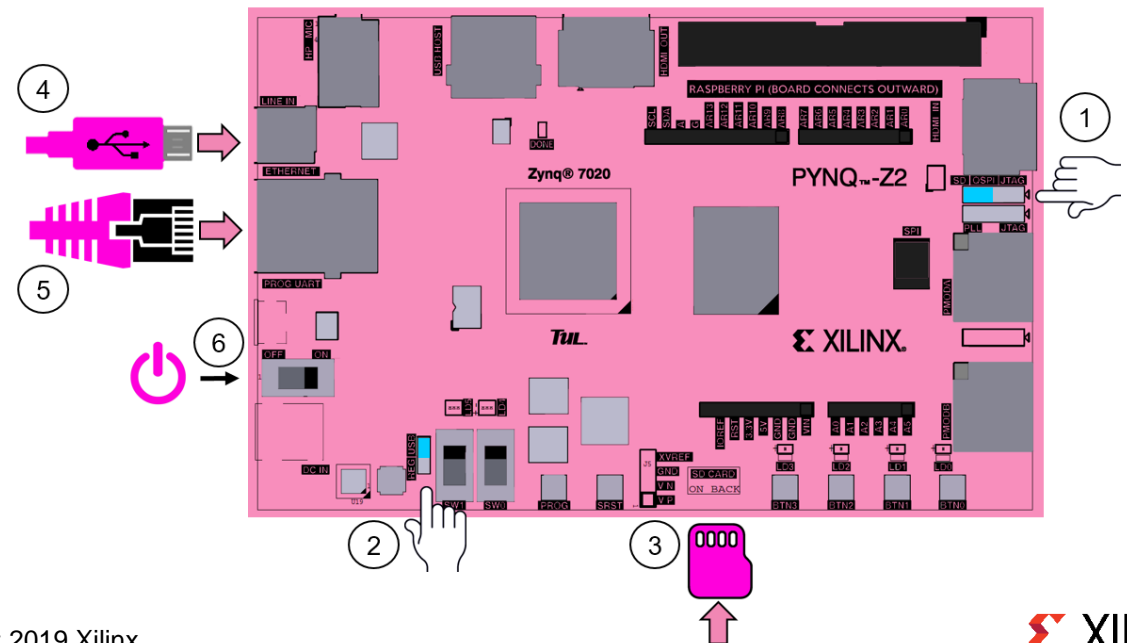
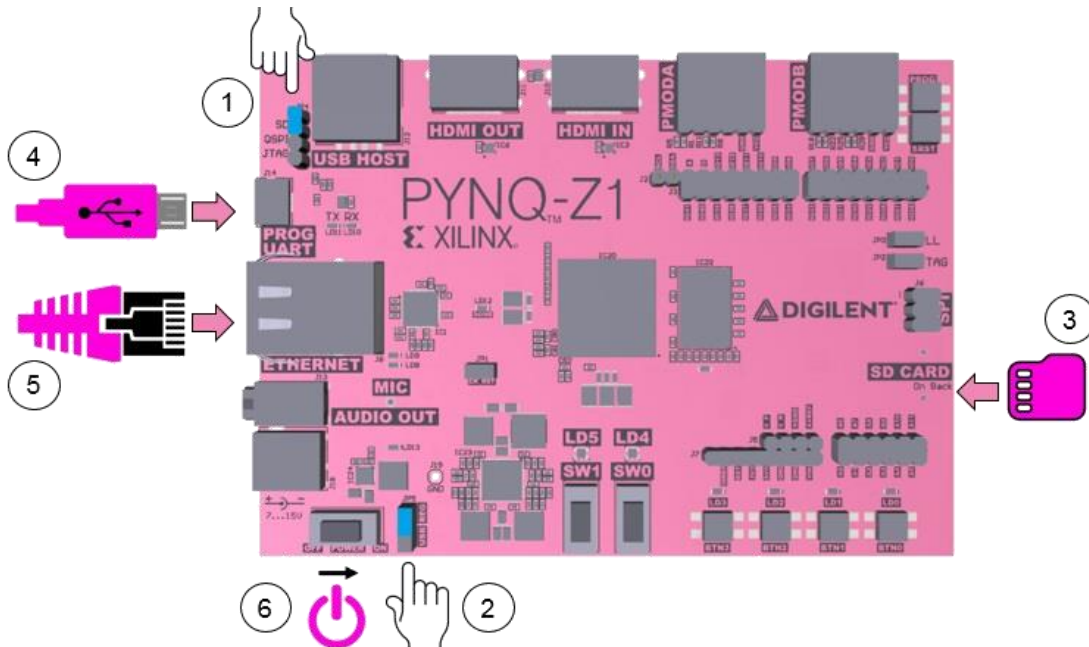
*Instructions to download and prepare SD card:
http://pynq.readthedocs.io/en/latest/getting_started.html

Pynq overview (PYNQ-Z1/PYNQ-Z2)



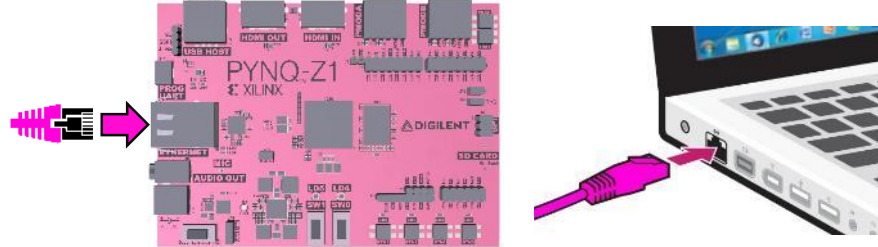
Connecting to the board

1. Configure board to boot from SD Card
2. Set Jumper to power from USB
3. Insert SD Card
4. Connect USB cable
5. Connect Ethernet cable to PC or to a Switch/Router
6. Power On

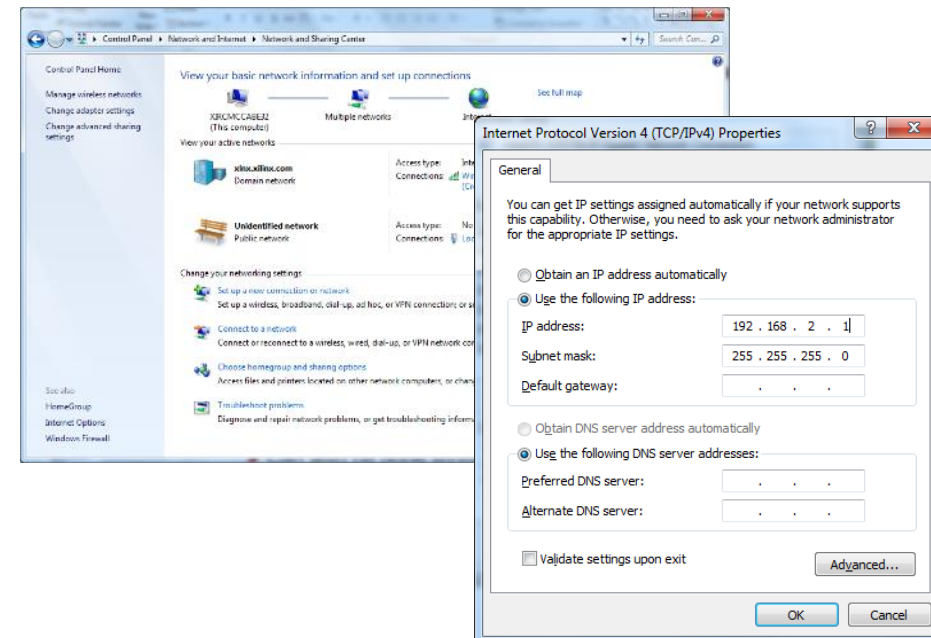


Connecting to the board – Direct connection

- > **Connect board directly to Ethernet port on PC**
 - >> USB to Ethernet adapter if no Ethernet port available
- > **Board IP will default to 192.168.2.99**
- > **Manually specify static IP for PC**
 - >> Must be in same range as board:
 - E.g. 192.168.2.1
 - >> No internet access unless you bridge network connections



Connect board directly to PC



Samba share

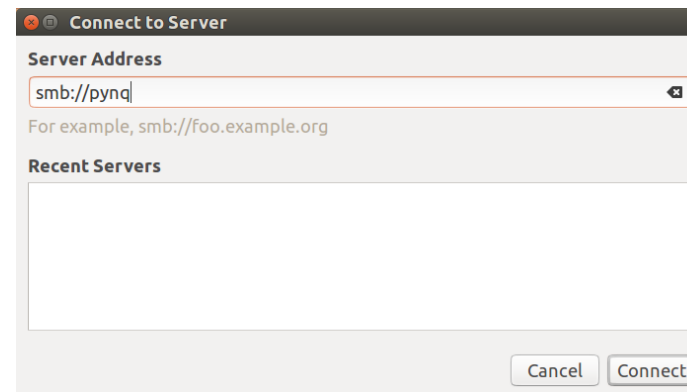
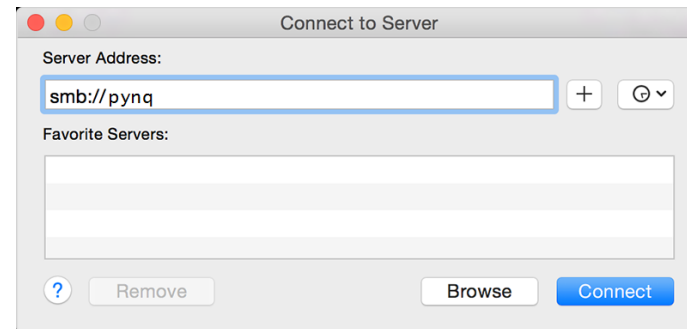
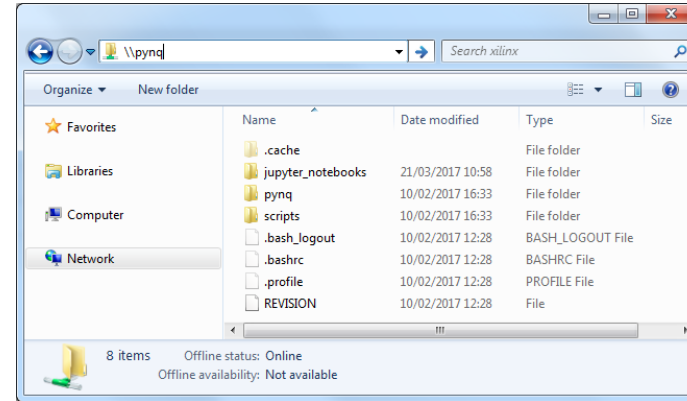
> Board can also be accessed as a shared drive

- >> Windows: \\192.168.2.99
- >> Linux: smb://192.168.2.99
- >> MAC OS: smb://pynq
 - Hit Command+K to bring up the 'Connect to Server' window

> Log-in

- >> Username = xilinx
- >> Password = xilinx

> Copy files easily between PC and board

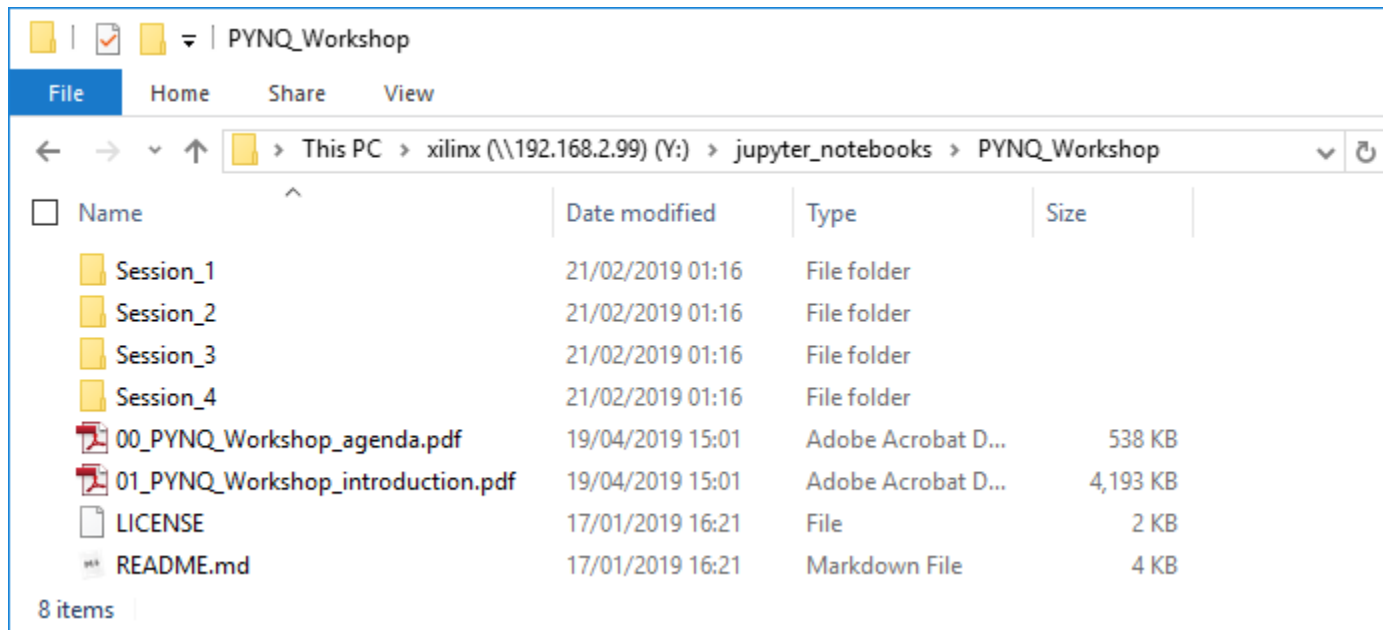


Introduction to Jupyter notebooks

➤ Copy the lab files into the folder:

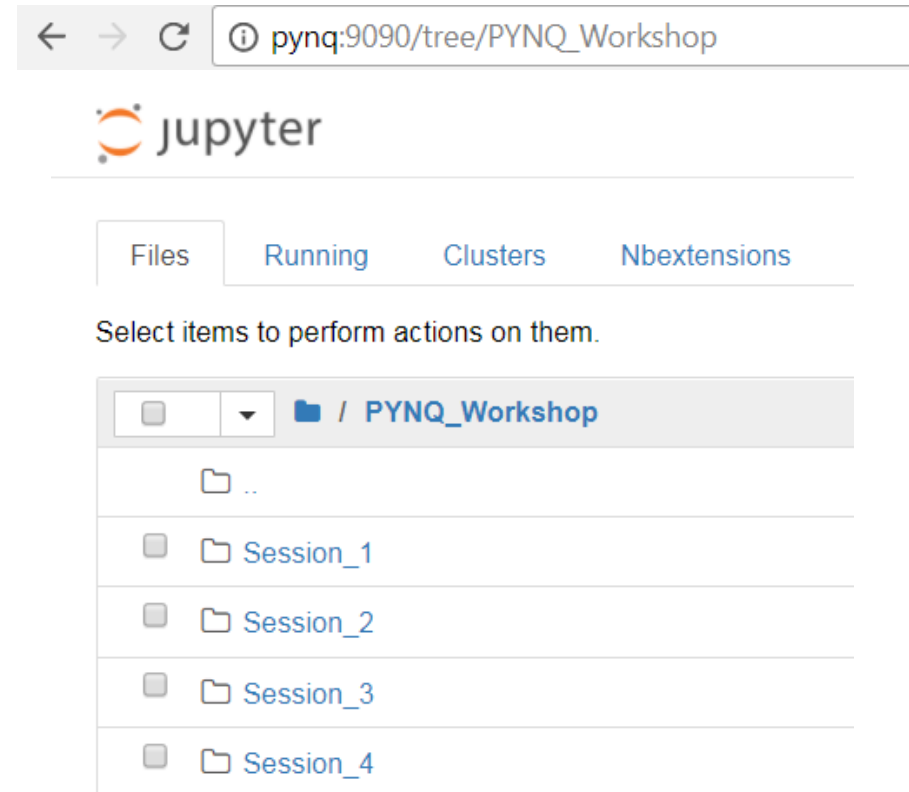
\\192.168.2.99\xilinx\jupyter_notebooks

>> If you don't copy the files into the jupyter_notebooks folder, you won't be able to see them in Jupyter



Log in to Jupyter portal

- > **Open a browser***
 - >> Chrome (preferred due to authentication issue on other browsers)
- > **Browse to: <http://192.168.2.99>**
- > **password = xilinx**
- > **Browse to *workshop* folder**



*<http://jupyter-notebook.readthedocs.io/en/latest/notebook.html#browser-compatibility>

** The PYNQ-Z1/Z2 boards don't have a realtime clock. The first time a board is used, the date and time of the system may be too far out of sync, and cause some browser (e.g. FireFox) to refuse setting a cookie which prevents log-in to the board. Chrome does not have this issue. To resolve this issue, update the date on the board. In a terminal execute: `sudo date +%Y%m%d -s "20180920" "20180920"` is YYYYMMDD

Lab exercises: Session 1 (1)

> **Getting started with Jupyter Notebooks**

- >> Notebook's browser-based interface
- >> Writing text with Markdown
- >> Writing and running Python scripts
- >> The IPython interpreter

> **Getting started with IPython**

- >> Executing OS shell commands
- >> The Ipython magic commands



Lab exercises: Session 1 (2)

> Exploring the board

- >> Getting CPU information
- >> Getting network status

> Programming on-board peripherals

- >> Controlling on-board LEDs
- >> Interacting with buttons, switches, and LEDs



Lab Review

> Jupyter Notebook

- >> Web application/server
- >> Create and share documents
 - Live code, equations, visualizations, explanatory text/comments, results

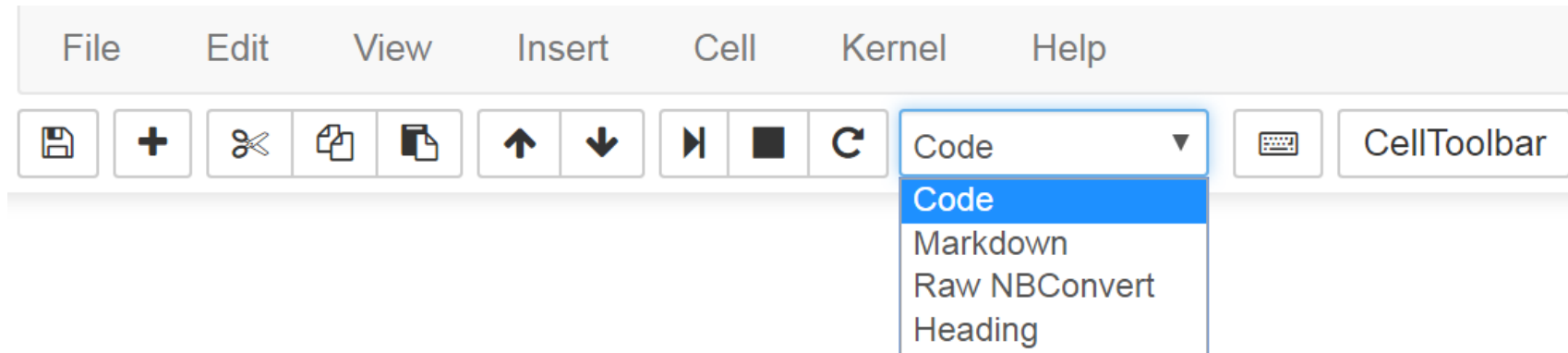
> Make a copy of notebook before editing

> Sharing Notebooks

- >> Save the iPython Notebook .ipynb
- >> Save as... HTML to share results



Notebook Cells



- > **Code**

- >> Python

- > **Markdown**

- >> Comments and notes

- > **Execute cells to run code/render**

- > **Split and reorder cells**

- > **Output printed after cell**

```
In [1]: import time
        for i in range (1,9):
            print(i*i)
            time.sleep(0.5)
```

```
1
4
9
16
25
36
49
64
```

Markdown cells

> # Headings

> * *Italics* *

> ** **bold** **

- * Bullet points

- * Bullet points

> <html></html>

> Not just comments. Explanatory text, notes, documentation.

Notes, comments and Markdown

You can write notes and comments using the `Markdown Language`.

This cell is a Markdown cell. Double click it now to see the raw markdown.

This is **bold**, this is *italic* and you can see how headings and sub-headings are indicated above using "#". You can also use `html` in markdown.

Execute the cell to render the markdown.

* Note in the dropdown box in the toolbar that this is a Markdown cell.

Shell commands, Cell magics

- > Execute shell commands directly from notebook
- > Prefix !
- > Built in Cell magics

```
!uname -a  
!whoami  
!pwd  
!ls  
!ping www.xilinx.com
```

```
%lsmagic
```

Available line magics:

```
%alias %alias_magic %autocall %automagic %autosave %bookmark %cat %cd %  
clear %colors %config %connect_info %cp %debug %dhist %dirs %doctest_mo  
de %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %lf  
%lk %ll %load %load_ext %loadpy %logoff %logon %logstart %logstate %l  
ogstop %ls %lsmagic %lx %macro %magic %man %matplotlib %mkdir %more %  
mv %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinf %pinf2 %  
popd %pprint %precision %profile %prun %psearch %psource %pushd %pwd %  
pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %rep %re  
run %reset %reset_selective %rm %rmdir %run %save %sc %set_env %store  
%sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos  
%xdel %xmode
```

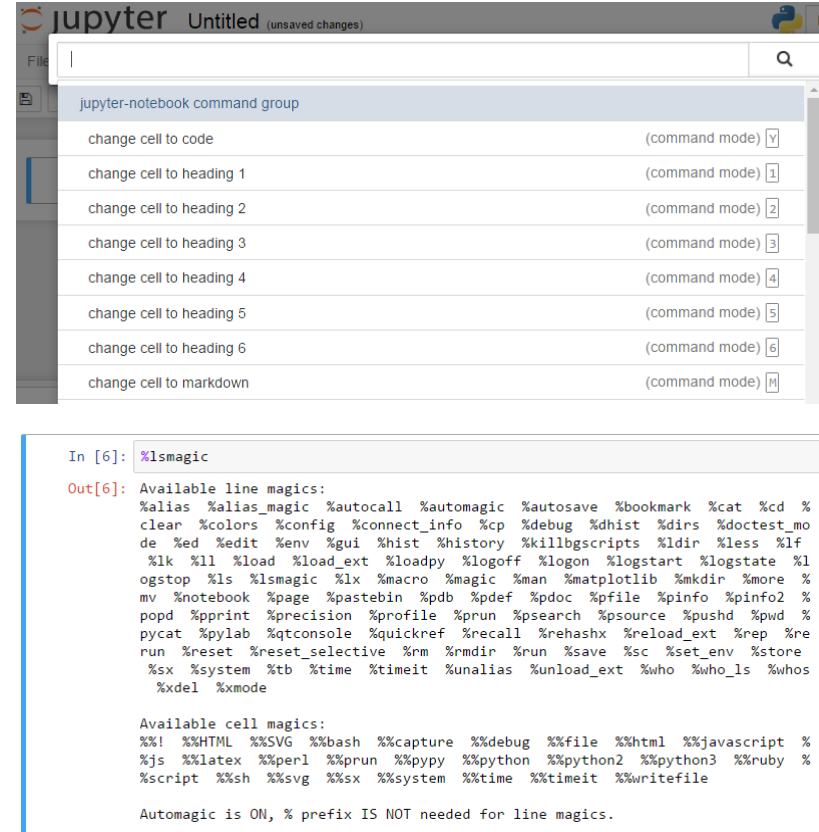
Available cell magics:

```
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript %  
%%js %%latex %%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %  
%%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile
```

Automagic is ON, % prefix IS NOT needed for line magics.

Help

- > **Command palette: Ctrl + Shift + p keys**
- > **Python help:**
 - >> Execute “?” In a cell
 - >> `help()`
- > **%magic – learn about Ipython magics**
 - >> `%lsmagic` – get list of magics
 - >> `%quickref`



The screenshot shows the Jupyter Notebook interface. At the top, the title bar says "Jupyter Untitled (unsaved changes)". Below the title bar, there is a search bar and a dropdown menu. The dropdown menu is open, showing a list of commands under the heading "jupyter-notebook command group". The commands are:

- change cell to code (command mode) [v]
- change cell to heading 1 (command mode) [1]
- change cell to heading 2 (command mode) [2]
- change cell to heading 3 (command mode) [3]
- change cell to heading 4 (command mode) [4]
- change cell to heading 5 (command mode) [5]
- change cell to heading 6 (command mode) [6]
- change cell to markdown (command mode) [H]

Below the dropdown menu, there is a code cell. The input is:

```
In [6]: %lsmagic
```

The output is:

```
Out[6]: Available line magics:
%alias %alias_magic %autocall %automagic %autosave %bookmark %cat %cd %
clear %colors %config %connect_info %cp %debug %dhist %dirs %doctest_mo
de %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %lf
%lk %ll %load %load_ext %loadpy %logoff %logon %logstart %logstate %l
ogstop %ls %lsmagic %lx %macro %magic %man %matplotlib %mkdir %more %
mv %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %
popd %pprint %precision %profile %prun %psearch %psource %pushd %pwd %
pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %rep %re
run %reset %reset_selective %rm %rmdir %run %save %sc %set_env %store
%sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos
%xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript %
%js %%latex %%perl %%prun %%pypy %%python %%python2 %%python3 %%ruby %
%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

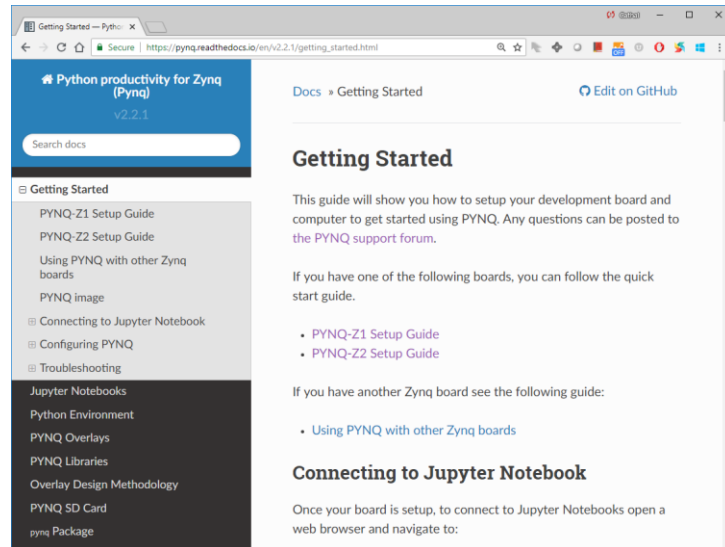

Documentation and Support

> Documentation

>> pynq.readthedocs.io

> Support

>> pynq.io/support



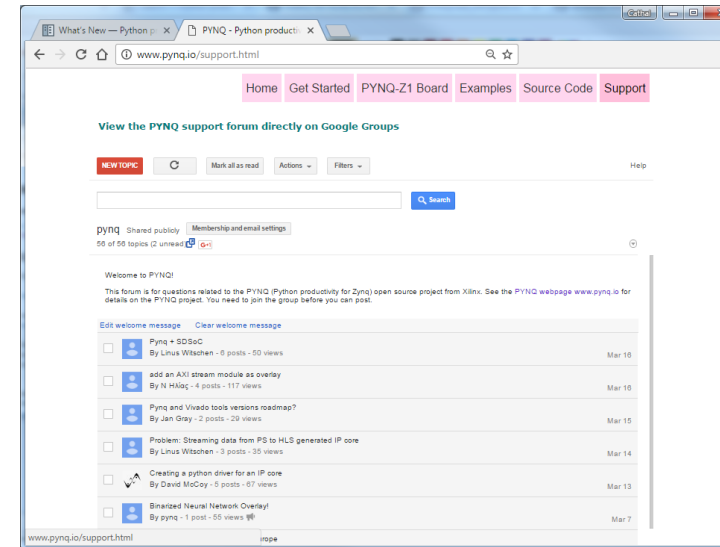
> GitHub

>> Issue tracker

>> github.com/Xilinx/PYNQ

> All accessible from

>> www.pynq.io



Next steps

> Examples

- >> How to use overlays
- >> Peripherals, Grove, Pmod
- >> Other peripherals, PS/PL
- >> Logictools



Questions?



Troubleshooting: LEDs, serial connection

> Status LEDs

- >> Power On: Red LED
 - Check the power source jumper is correctly set
- >> Bitstream Loaded (Pynq booting): Green “Done” LED
 - Make sure that the boot source is set to SD card and the SD card is inserted

> Serial connection to the board

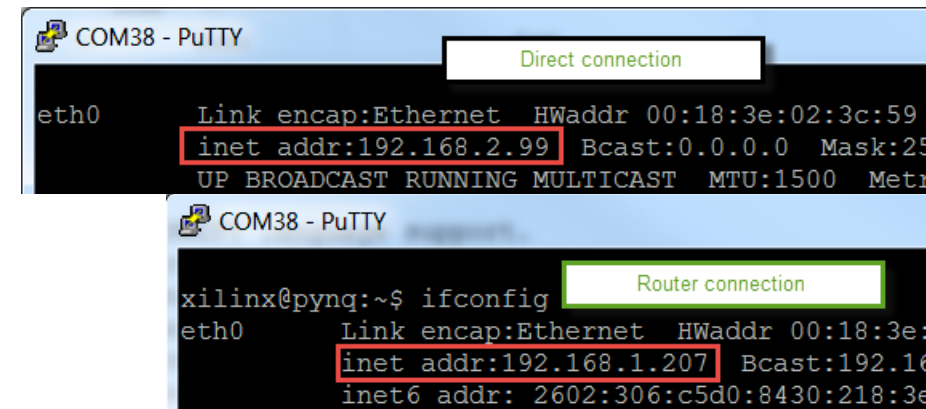
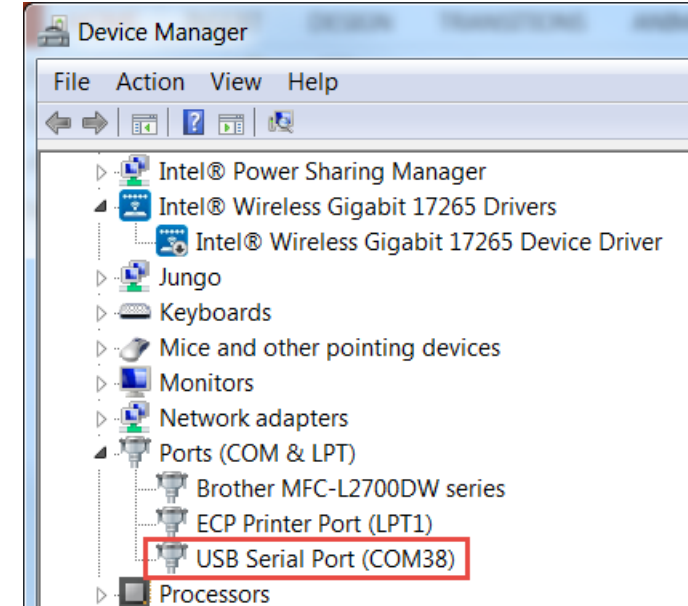
- >> 115200 baud (data:8 bit, stop bits:1)
- >> Windows: Device manager to get COM port
- >> Check board has booted
 - Is Linux shell available on terminal?

> Web browser connection

- >> Check/Modify board IP
- >> *ifconfig* to check network settings

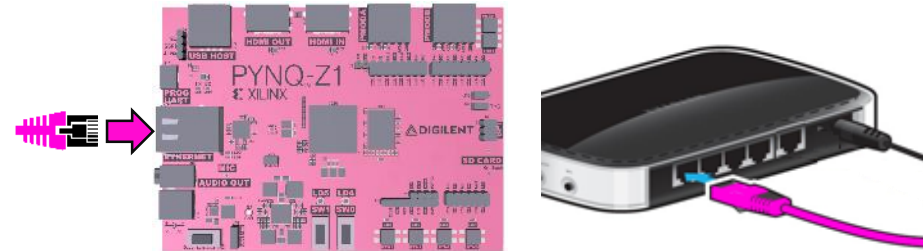
> Clearing browser cache

- >> Shift+Ctrl+Del

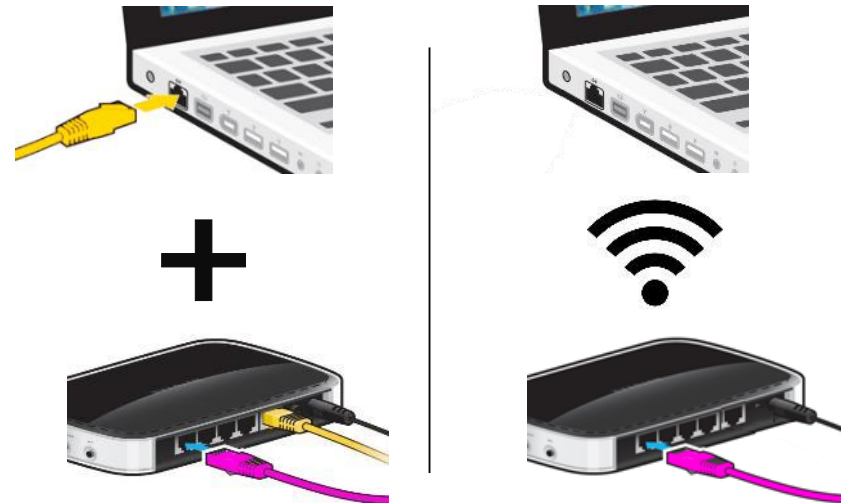


Connecting to the board – Via network switch

- > **Board is connected to network (switch/router)**
 - >> If DHCP server running on network board will get IP from DHCP server
- > **Internet access via network allows Python packages to be installed/updated**
- > **PC can connect to same network**
 - >> Wired
 - >> WiFi



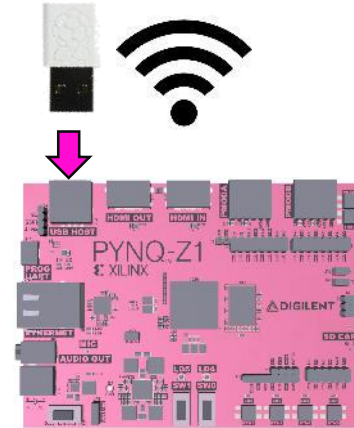
① Connect board to switch/router



② PC connected to router via cable or wireless

Connecting to the board – Via WiFi

- > **Connect USB WiFi dongle to board**
- > **Connect board and PC to wireless hotspot**
 - >> Router, Mobile Phone
- > **Need to configure board to connect to WiFi network**
 - >> Notebook available to do configuration
 - `usb_wifi.ipynb`
 - >> Manually configure from terminal
 - Modify `/etc/network/interfaces.d/`
 - `iwconfig` in next release



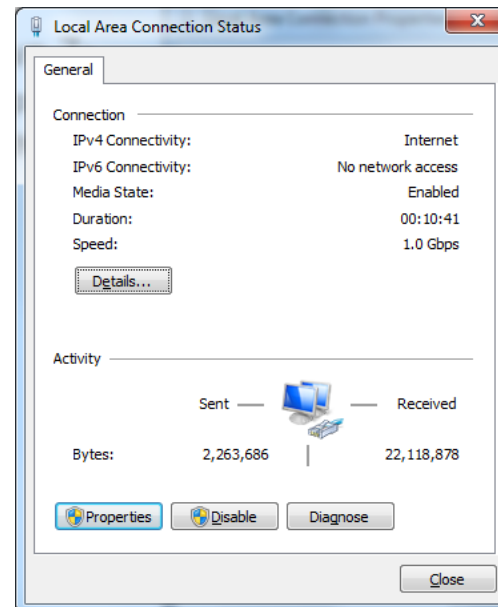
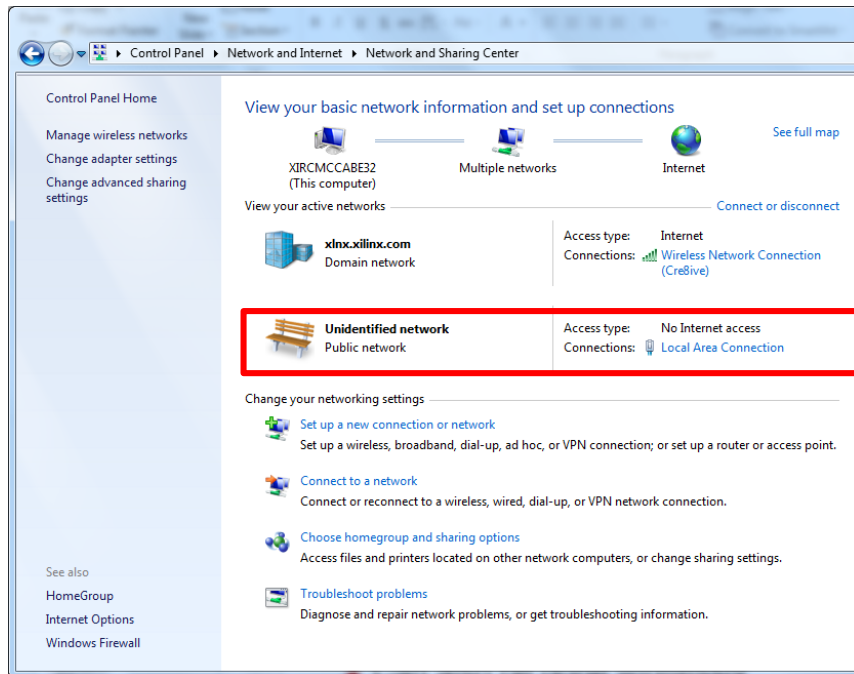
- ① Connect WiFi dongle to board
- ② Connect board and PC to wireless hotspot



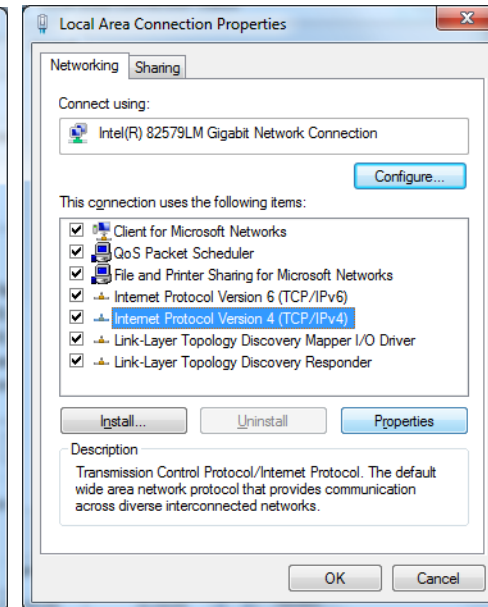
Configure Ethernet - Windows

> Windows 7:

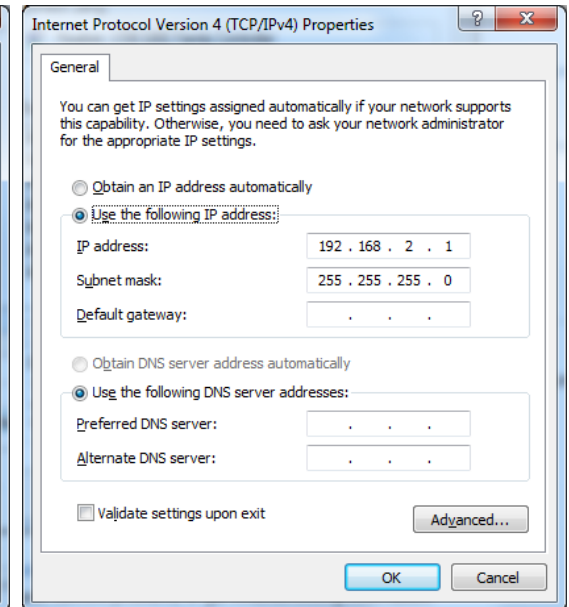
>> Control Panel\Network and Internet\Network and Sharing Center



Properties



TCP/IPv4→Properties

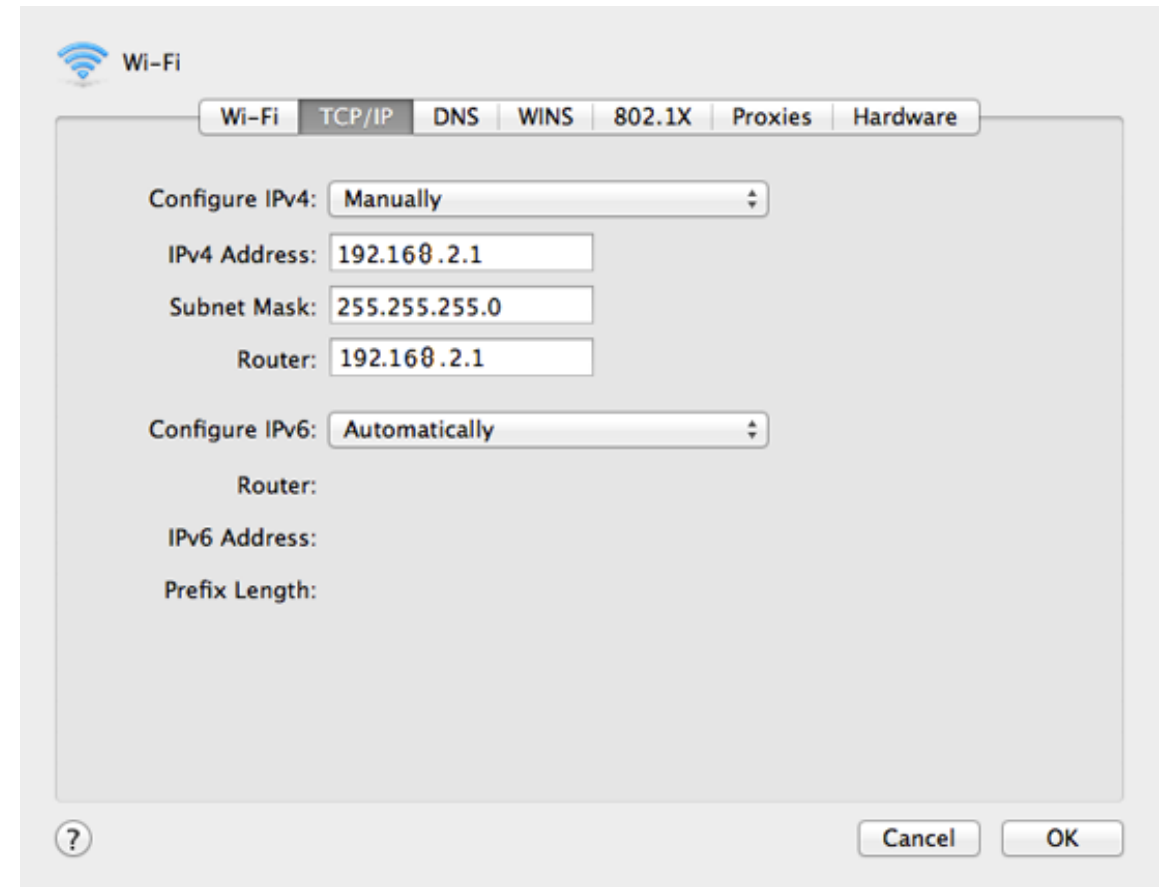


Set IP address

Configure Ethernet - Mac

> Mac OS:

- >> Apple menu > System Preferences > Network
- >> Select interface > Advanced
- >> Command + K



Adaptable.
Intelligent.

