1. Raspberry Pi getting started
2. Power on
   1. Connected to own HDMI monitor. Monitor says: no signal
   2. Removed the SD card and edited in config.txt to uncomment:
      1. # uncomment if you get no picture on HDMI for a default "safe" mode
      2. #hdmi\_safe=1
   3. Still no signal. Connected to LG TV HDMI input: Works!!
3. Ran the “first boot” sequence, assigned passwords and wifi connection.
4. Inspected my monitor: turns out that the monitor could not automatically switch to the HDMI input. After selecting HDMI 1 input by the side pushbuttons **SUCCESS!**.
5. WiFi: Connected during the first boot sequence without problems.
   1. Assigned permanent IP 192.168.1.170 in my router.
6. VNC connection: the client is started easiest by opening the vnc://<IP> url in a browser. The screen sharing built in app says version mismatch: “The software on the remote computer appears to be incompatible with this version of Screen Sharing.”
   1. Started the “vncserverui” app on the RP, and changed the option Security.Authentication from “Unix password” to “VNC password”. Set the password to same as system. **SUCCESS!**
7. Samba server mounting:
   * 1. Followed the guide. SUCCESS
     2. So now I have a working RP400 on remote desktop from my Mac.  
        SSH access by ssh pi@192.168.1.170 OK.
8. Remote desktop
   1. Note that in the RP400 must have a HDMI monito connected in order to start the graphical desktop. I connect it permanently to a HDMI input in my monitor, even when controlling via en remote desktop.
   2. The Mac has a built in “desktop sharing” app that supports VNC. The RP400 VNC server must be enabled in Settings, and the VNC server’s authentication method must be changed to “VNC password”. The desktop sharing app is hidden, and must be searched for in the Library folder. Connetion to the standard port worked without problems, but when reconencting later you have to use the “conenction” menu, ignoring the connection pop-up window.
   3. Now I have both an SSH mounted and remote desktop-accessed RP400.
9. Connecting the Pico and running Hello World
   1. Installed Thony, and followed the getting started guide to write hello world and blink LED programs in Python.
10. Installing the C/C++ IDE and SDKs
    1. Visual Studio from visualstudio.com. OK. Installed Python Jupiter, PHP and C/C++ extensions.
    2. Cloned the Pico SDK from Github
    3. Ran "blink" and "HelloWorld".
11. Running code on the Pico
    1. https://www.okdo.com/getting-started/get-started-with-raspberry-pi-pico-visual-studio-code/
12. Using the RP400 as a serial port master
    1. Use /dev/serial0 (GPIO pins 8, 10)
       1. In preferences.configuration.interfaces enable serial port and disable serial console.
       2. Connect Rx (GPIO 10) to Tx (GPIO 8)
       3. Install minicom
       4. run minicom -b 115200 -o -D /dev/serial0
       5. Verify that typing in the terminal is looped back into the screen.
       6. A python testing script:
          1. Write strings with serial number
13. Building a new project:
    1. Based on the uart\_advanced example
    2. Started a new workspace parallel to the pico\_examples workspace