Date: 25, Jan, 2023
Name: Mehrdad Zarei **Email: mehr.zarei1@gmail.com**

Current Project

This project is for automatically relocking the External Cavity Diode Lasers (ECDLs) by wavemeter and/or cavity transmission using Redpitaya.

Overview

In this project I am presenting a web application dedicated for Redpitaya board for automatically and real-time relocking the external-cavity diode lasers (ECDLs) by cavity transmission and/or wavemeter signals. Here specifically this web application is used in Strontium (Sr) Optical Tweezer Machine for quantum computing and simulation. But it can be used in different cold atom applications, such as optical atomic clocks, transportable optical clocks, future satellite missions, space missions, and wherever you want to fully control the ECDLs. Developing redpitaya has been considered in two parts, first FPGA developing but here we just show the requirements for developing because there is no need of changing the main bit stream file of fpga, secondly web-API developing including frontend and backend. At the end we have developed a web server on system of wavemeter for sending information of wavemeter and controlling remotely DigiLock and sending status to the clients (for example in this case redpitaya) also, I have explained how to relock ECDL by wavemeter and/or cavity transmission.

Progress

In this report you can find mostly information is needed for developing Red Pitaya. But before reading this report you have to read the main manual completely from these links:

https://redpitaya.com/rtd-iframe/?iframe=https://redpitaya.readthedocs.io/en/latest/quickStart/needs.html
https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn_fpga/fpga_learn.html

1- FPGA developing

For developing fpga program, recommended operating system appropriate for redpitaya at this time is Ubuntu 18.04, if you are not working with this OS, is recommended to install it on Oracle VM Virtual Box and set as much as possible more processors and storage like the figs 1-3 in the below. As you can see more than 24 GB memory and 8 processors has dedicated to Ubuntu. Now we need to install a software on Ubuntu to be able to write fpga program or modify initial fpga program for redpitaya. One of these software for this purpose which is used by RedPitaya team also is Vivado 2020.1 from Xilinx. All details for installation is written in the link below:

https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn_fpga/3_vivado_env/top.html

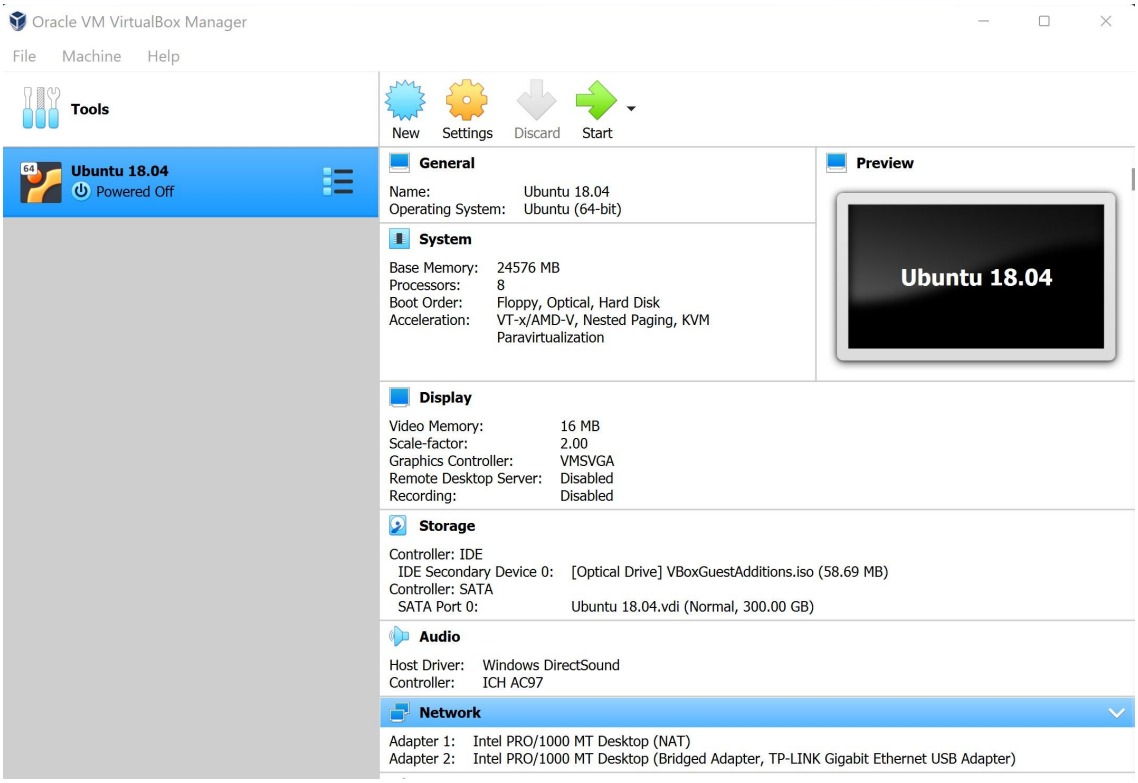

Fig. 1. Installed Ubuntu on VM.

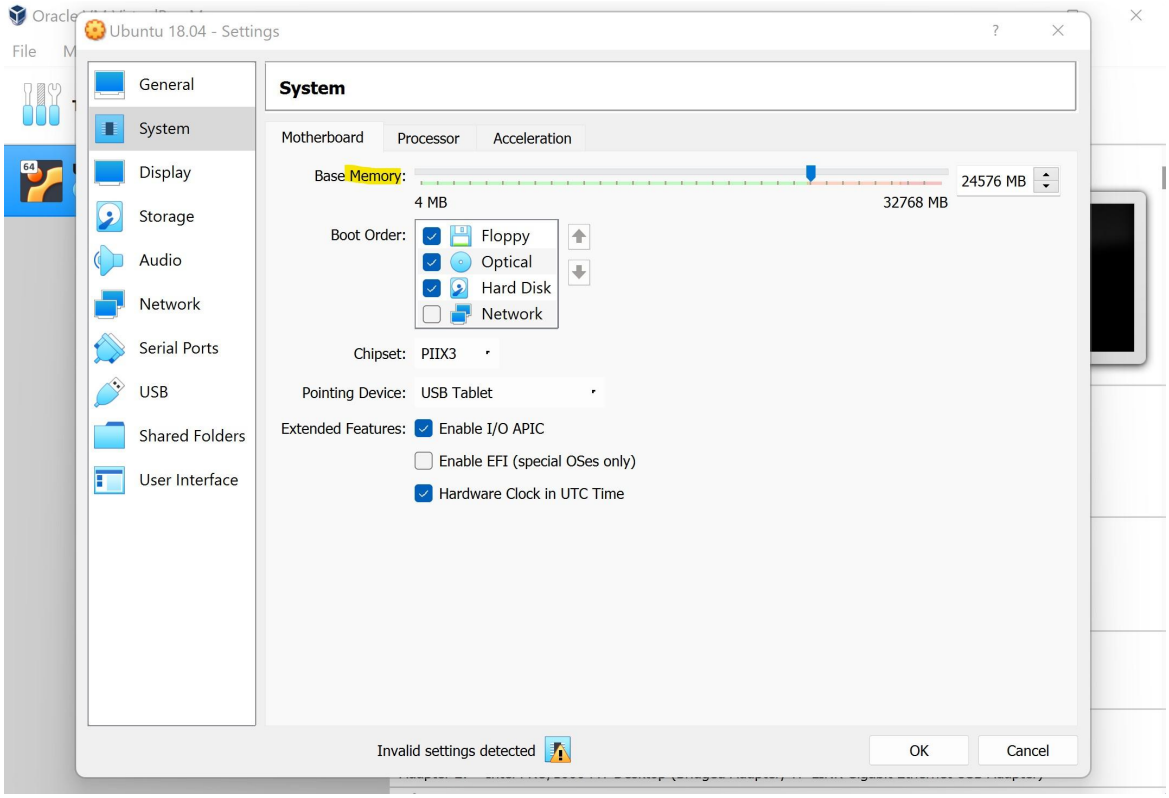

Fig. 2. Dedicated memory for Ubuntu.

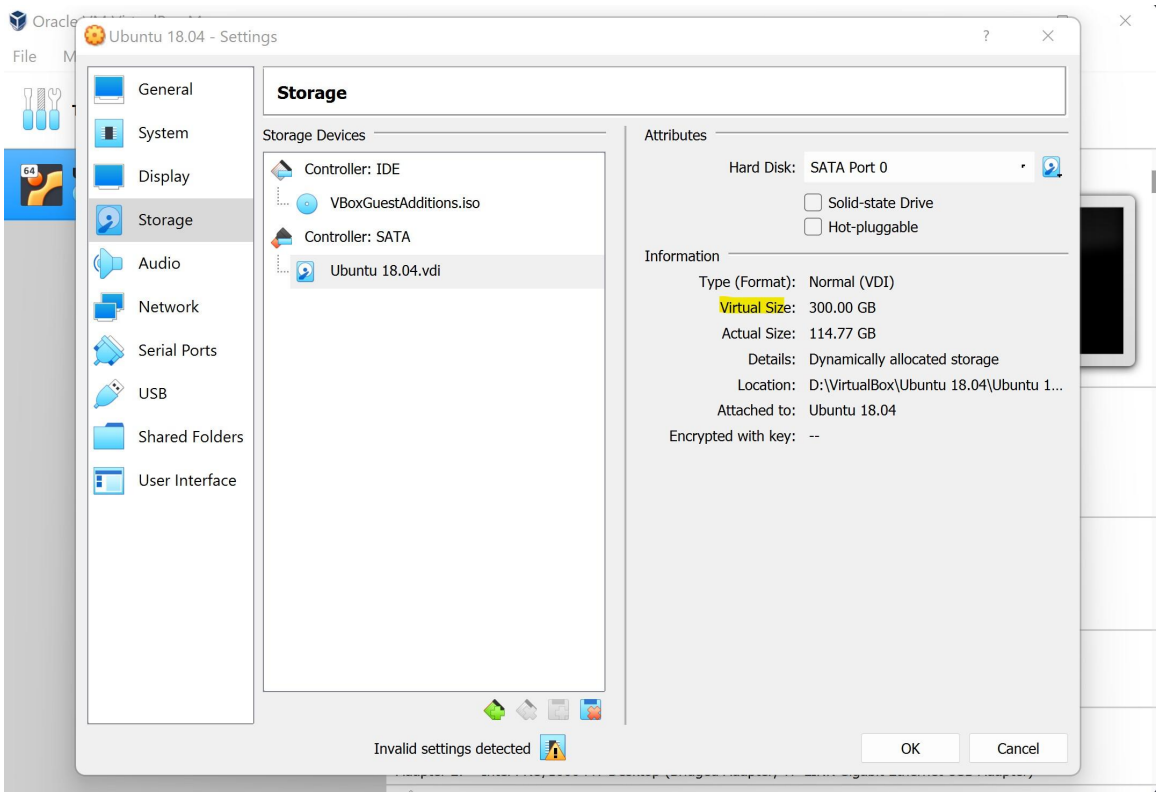

Fig. 3. Dedicated storage for Ubuntu.

After successfully installation of Vivado, you should clone fpga GitHub repository from this link:

https://github.com/RedPitaya/RedPitaya

check the fpga folder as fig. 4, if you don't have last stable version of fpga program (at this moment is v0.94) you should find another branch for cloning as you can see in the fig. 5. Then you should be able to open the initial fpga program for modifying or adding your own modules. To run the Vivado you should execute this command first:

. /opt/Xilinx/Vivado/2020.1/settings64.sh

After installing Vivado check that this file (settings64.sh) is exist. If you want to just run Vivado without initial fpga program of redpitaya you can execute next command:

vivado

If everything is fine vivado will start like fig. 6.
Note: maybe in your case Vivado is not installed in opt folder during installation process!
For opening initial fpga program at first go to fpga folder of cloned folder then execute this command:

make project PRJ=v0.94 MODEL=Z10

If everything is fine initial project of version 0.94 will be open like fig. 7.
Note: if your project is not in the home directory, you should have access to root then execute upper commands. To have access the root you should run this command: sudo su

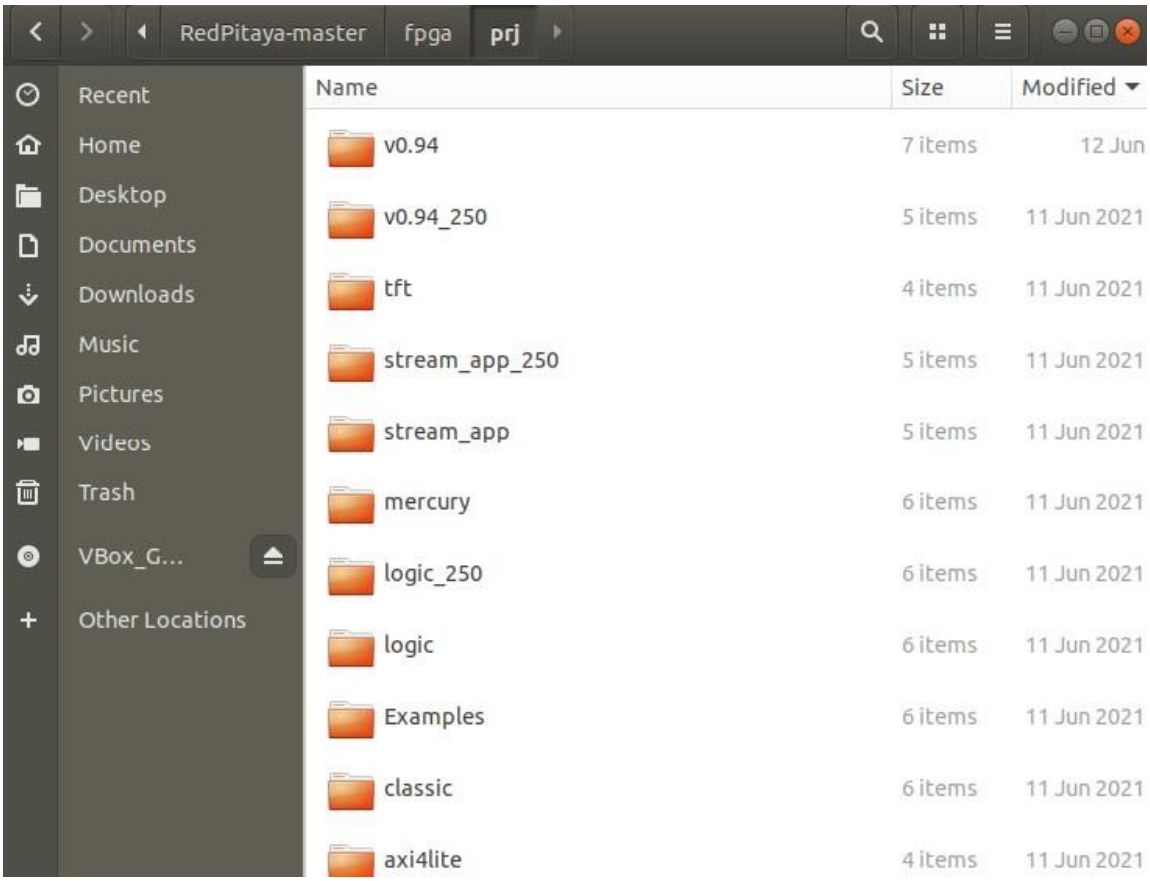

Fig. 4. Last stable version of fpga program.

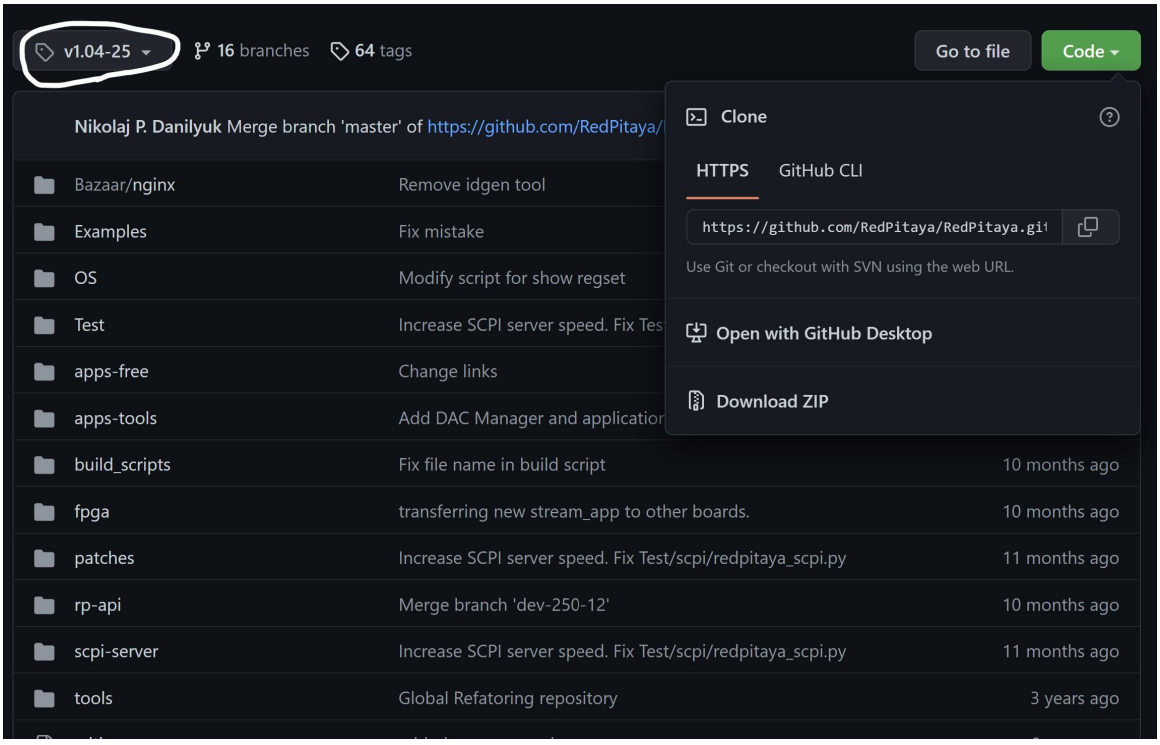

Fig. 5. Finding appropriate branch for cloning on GitHub.

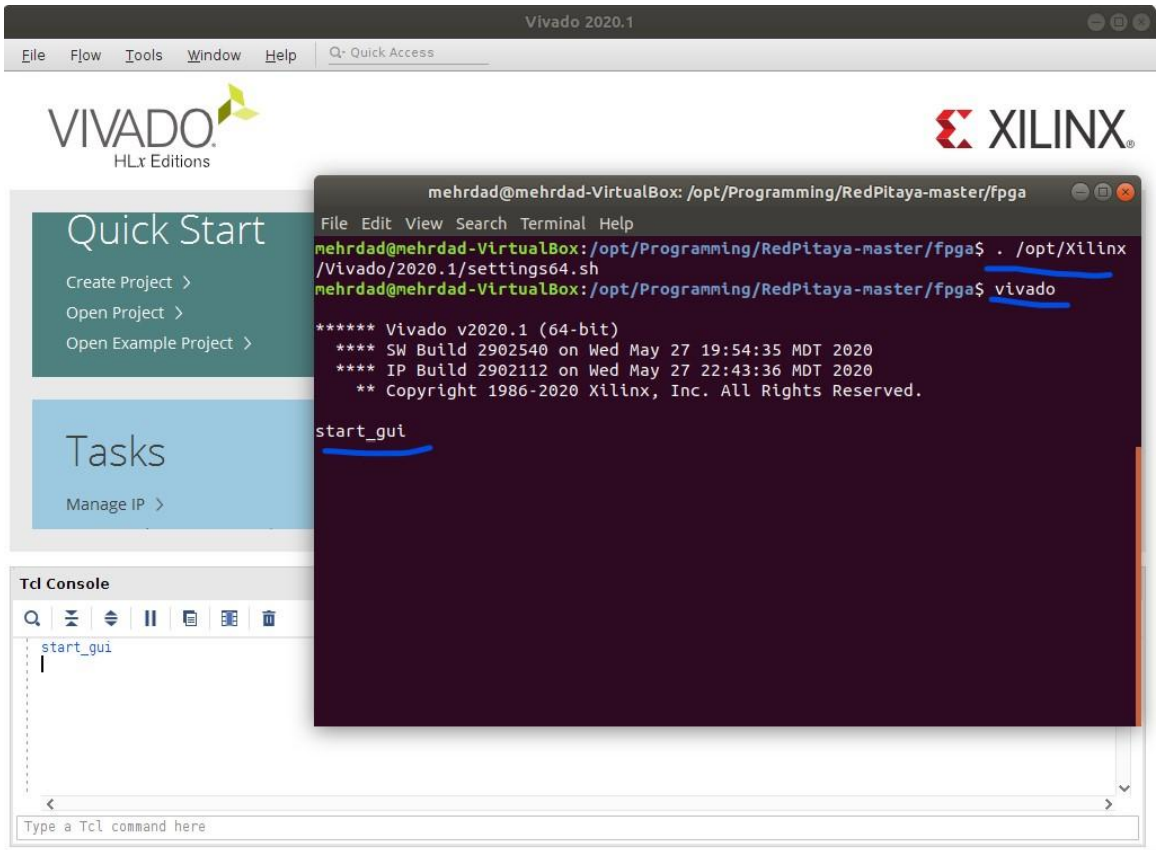

Fig. 6. Start Vivado on Ubuntu.

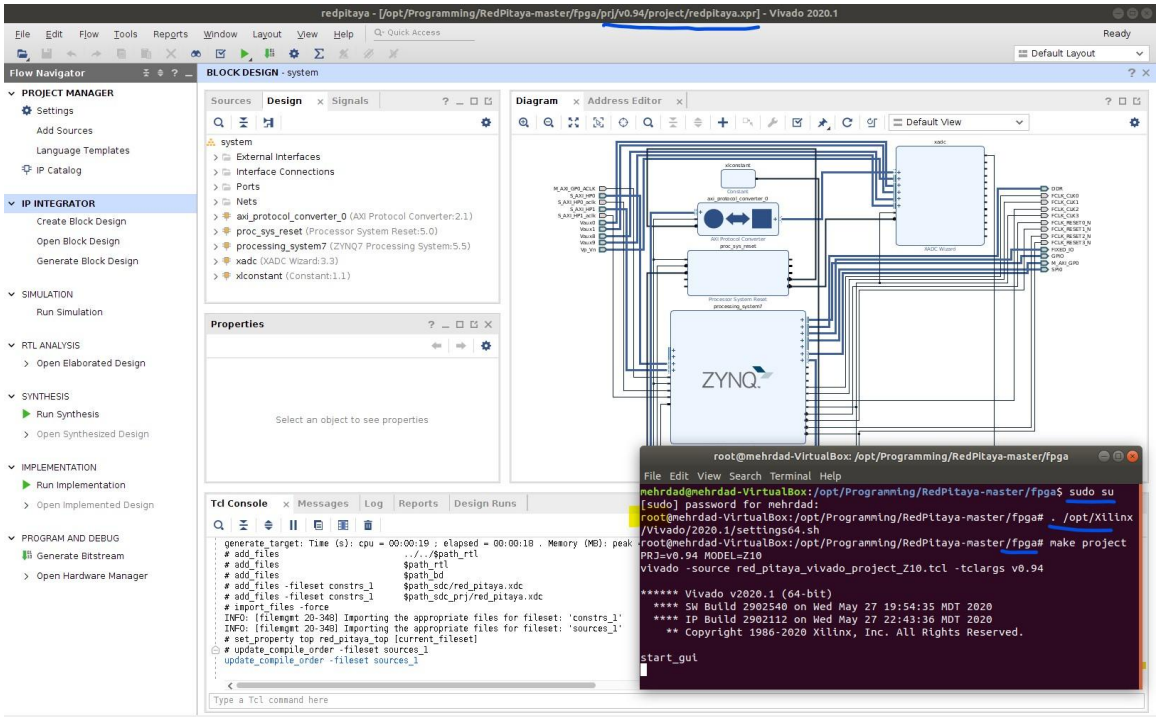

Fig. 7: Opening initial project on fpga of RedPitaya.

Now if you want to add your own code or module, you have to open red_pitaya_top.sv file from sources tab and modify the file and at the end we have to Synthesis, Implement and Write bitstream file then load this bitstream file on the fpga of the RedPitaya like fig. 8. For more information and how to do these process look at this link:

https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn_fpga/3_vivado_env/tutorfpga2.html

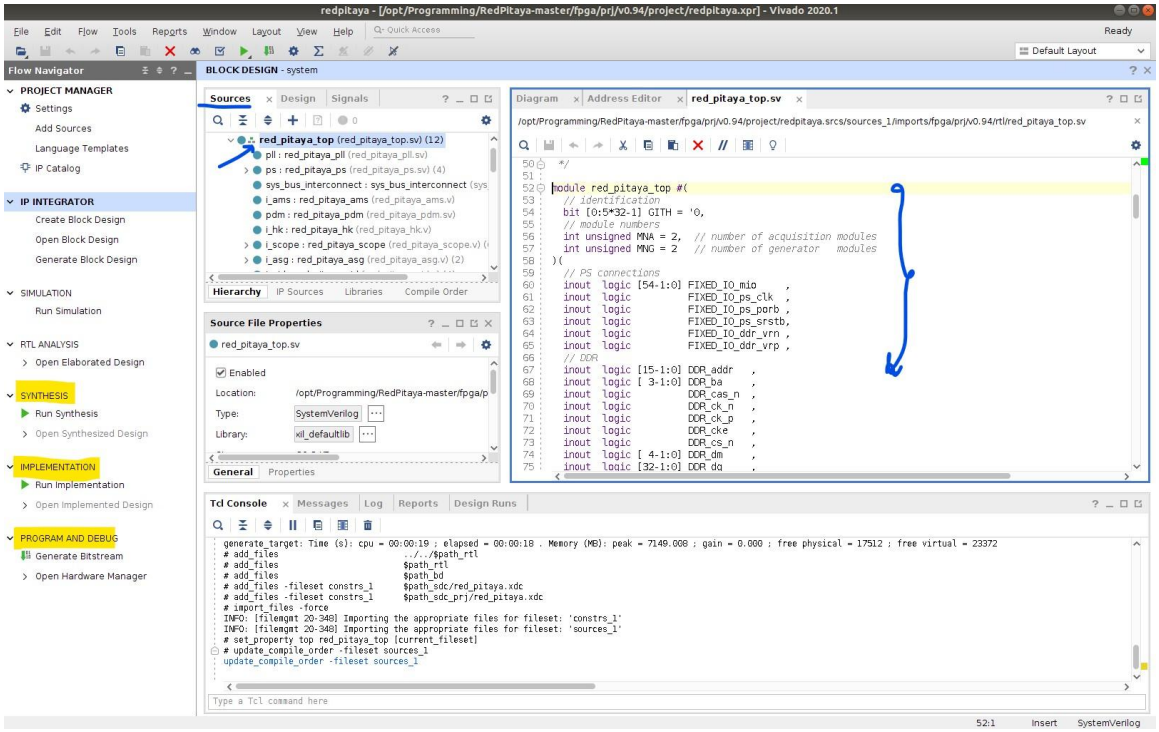

Fig. 8: Source code of initial project.

2- Web-API developing

Developing web-API on RedPitaya is divided by 2 parts: frontend and backend fig. 9.

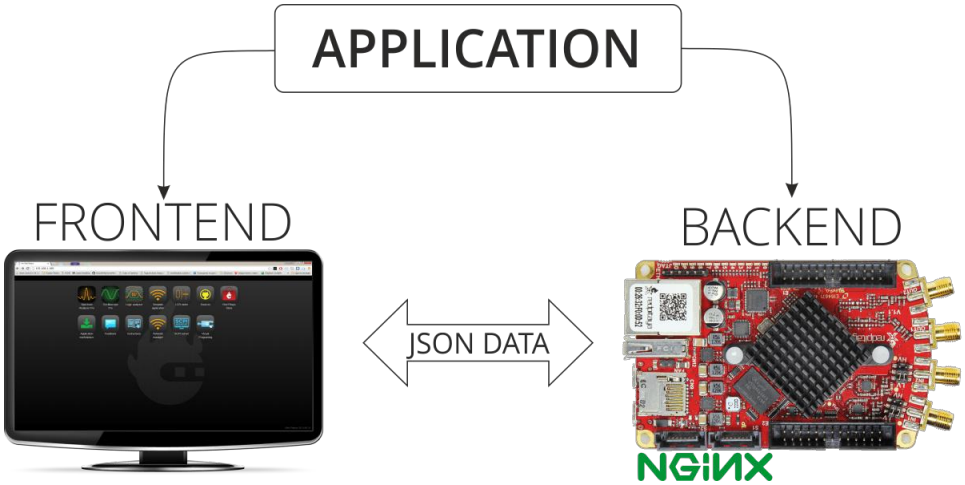

Fig. 9: Web-API developing.

Frontend is client side which user has interface with it, but backend is server side which is responsible to reply to clients and communicate with hardware. For developing frontend I used HTML5 for layout, CSS3 for element styles, and JavaScript for creating fast and reliable web application, but for developing backend I used C/C++. Frontend and backend process on RedPitaya have shown in figs. 10 – 11. For more information, examples, and how to do these process look at this link:

https://redpitaya.com/rtd-iframe/?iframe=https://redpitaya.readthedocs.io/en/latest/quickStart/needs.html

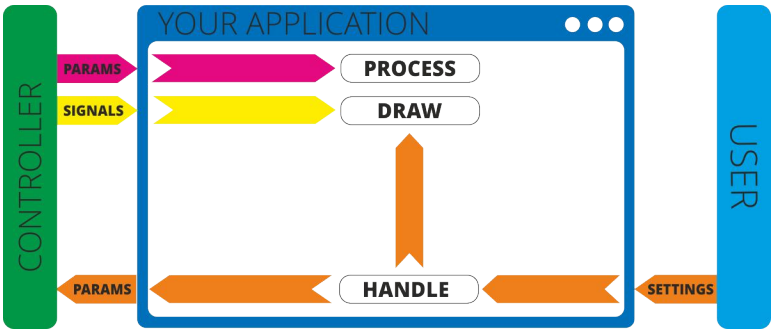

Fig. 10: Frontend process.

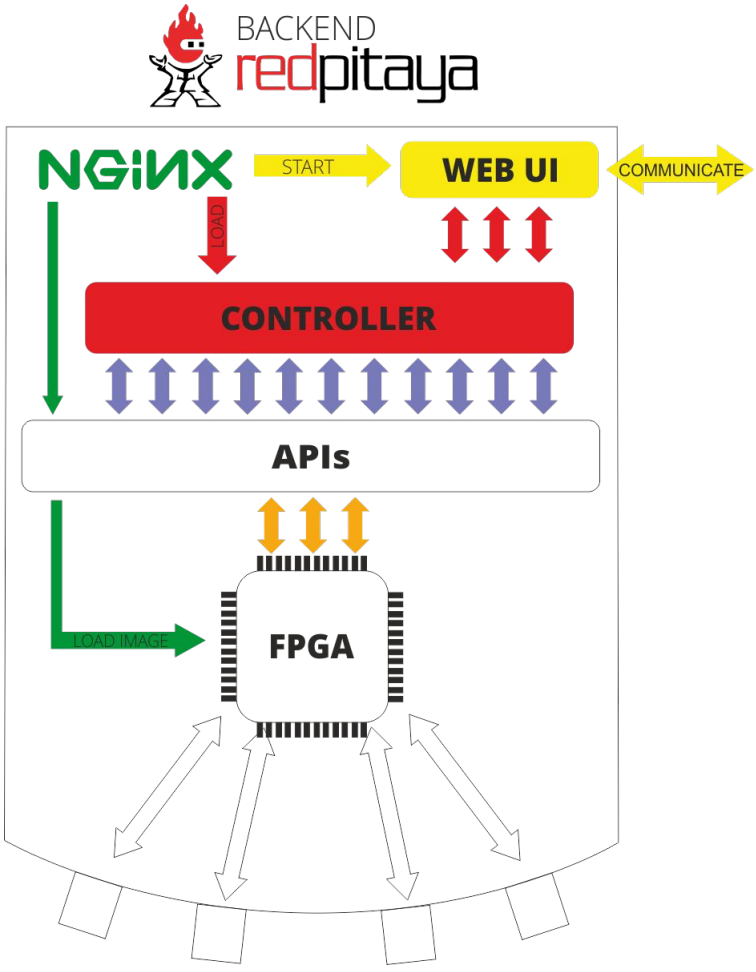

Fig. 11: Backend process on redpitaya.

In the fig. 12 you can see the developed application of relocking by wavemeter and cavity transmission.

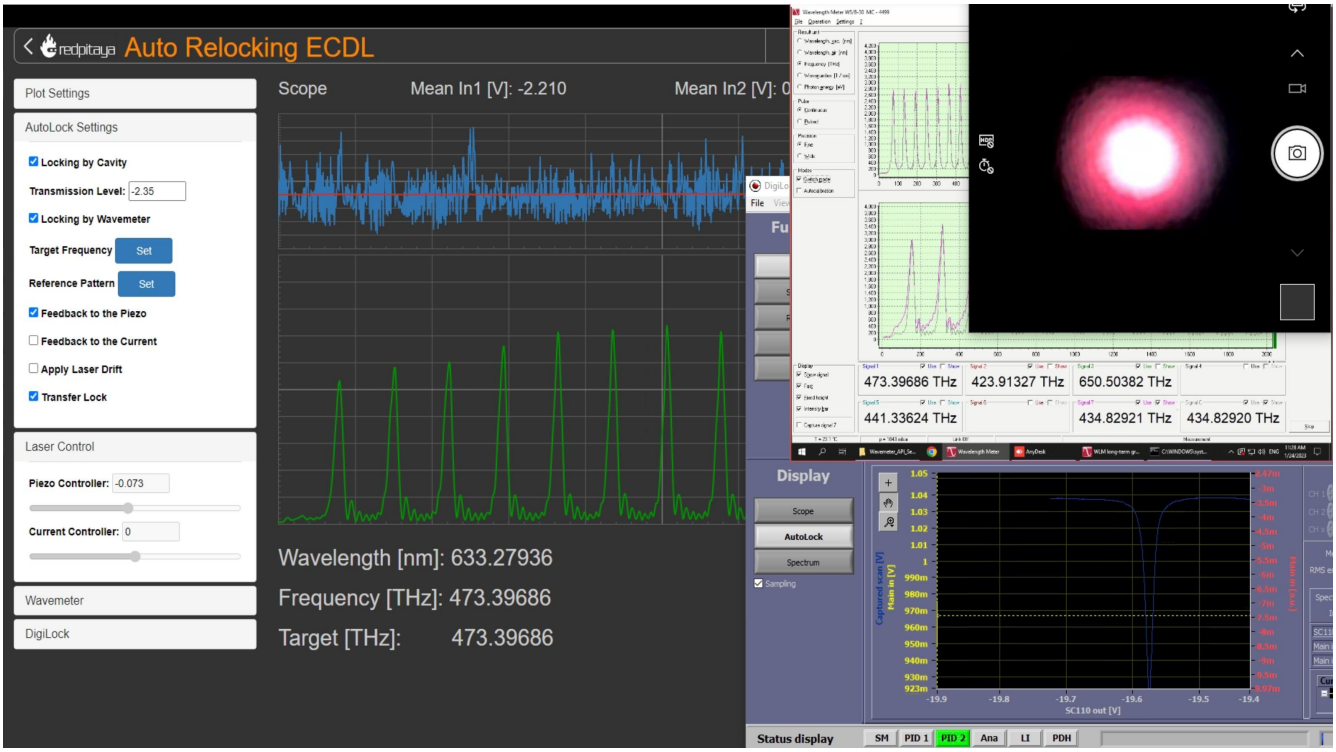

Fig. 12: Auto Relocking ECDL application by wavemeter and cavity transmission.

3- Web server for wavemeter

For relocking laser by wavemeter we have to get information of wavemeter like wavelength, frequency, and/or spectrum of laser light connected to wavemeter to be able to know that laser is locked (on target frequency) or not. Therefor I developed a web server by socket programming on python to reply requests of clients (RedPitaya, …) in fig. 13 whole communication is shown. So before running relocking application you should be sure that server is running otherwise you have to double click on run_wlmServer.bat file in wavemeter's system as shown in fig. 14.

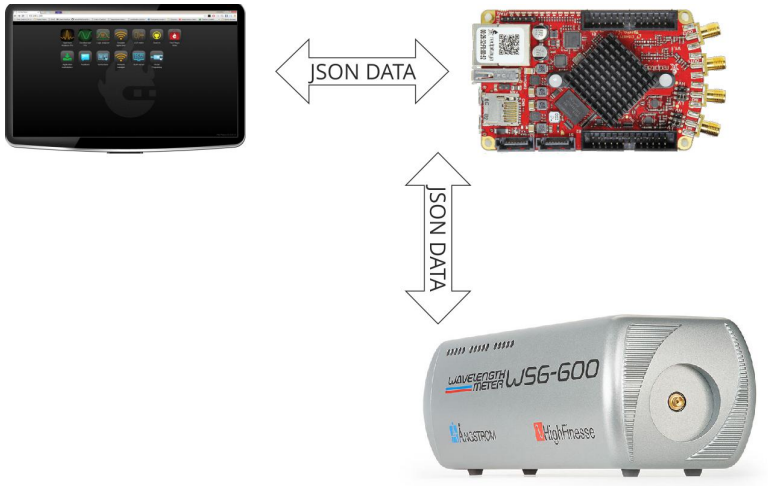

Fig. 13: Wavemeter and RedPitaya communication.

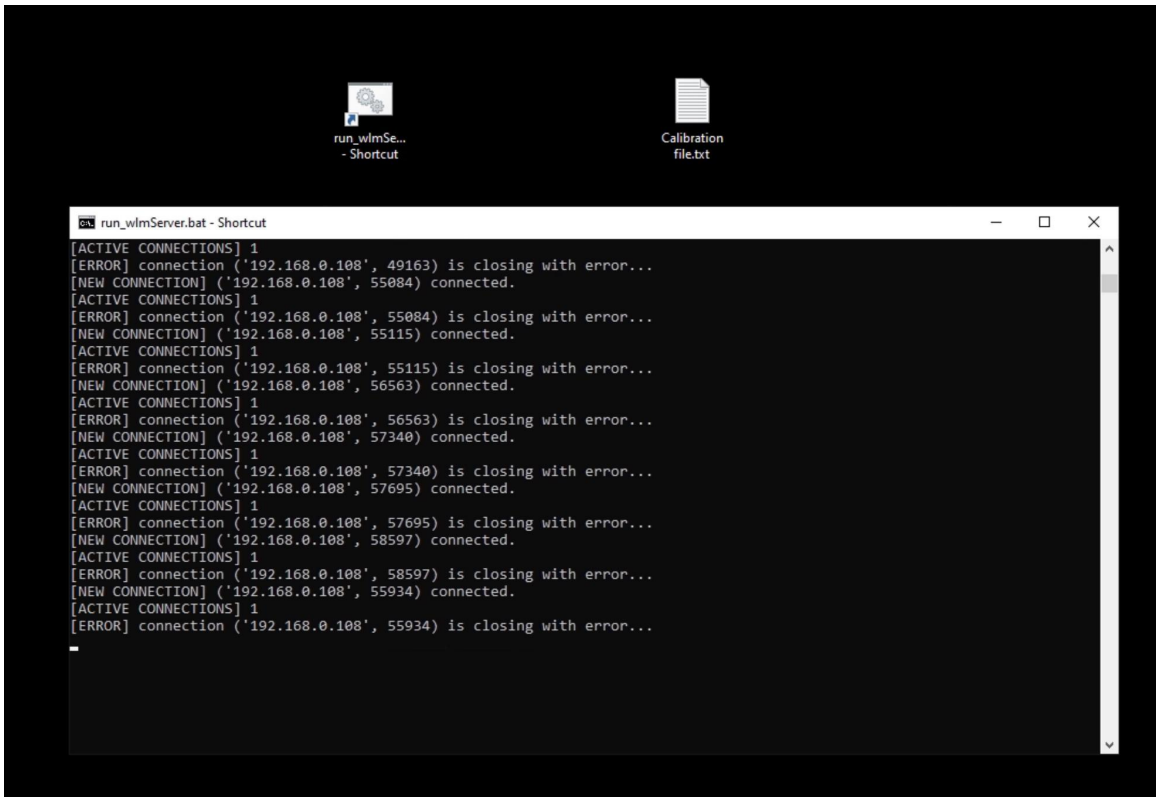

Fig. 14: Web server for wavemeter.

4- How to work with Relocking application

After preparing the redpitaya board and SD card, you should follow these steps:

A- Connection

First step is to connect the redpitaya to the network by router, WiFi, or directly to the PC. In the fig. 15 is shown router connection, and fig. 16 directly is connected to the PC, for WiFi you should use dongle.

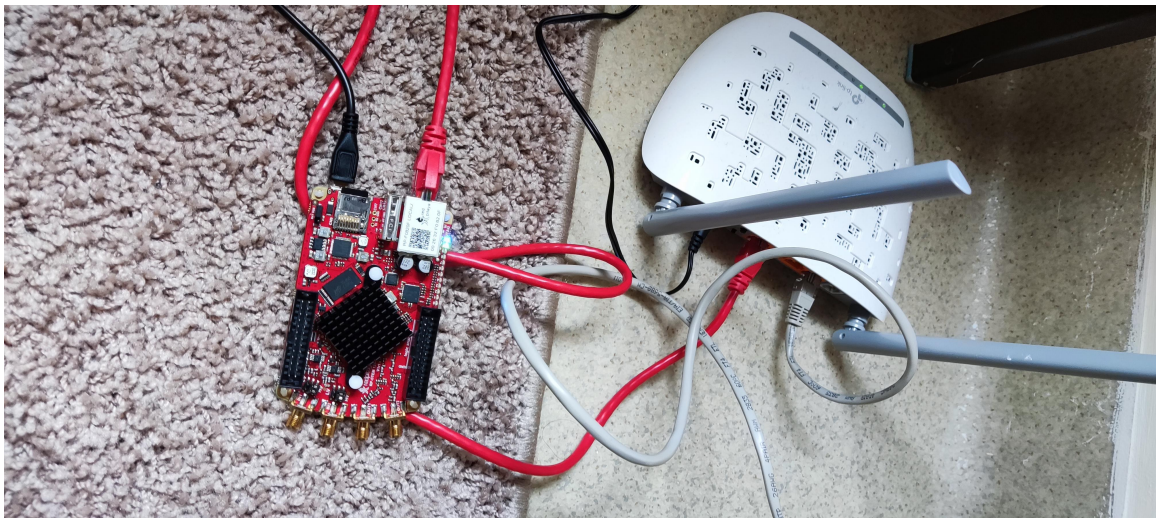

Fig. 15: RedPitaya connected to the router.

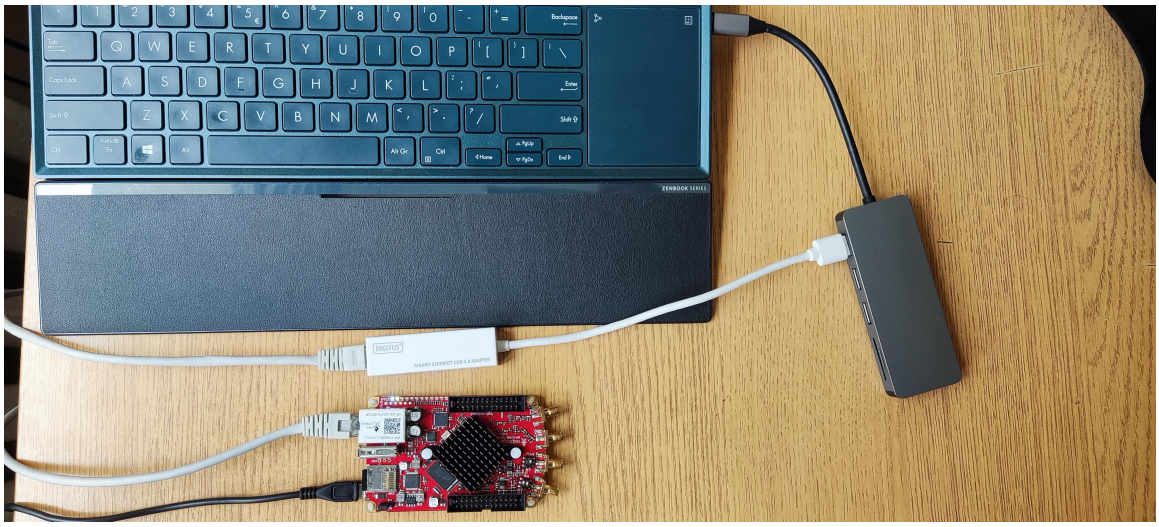

Fig. 16: RedPitaya directly connected to the PC.

After Ethernet connection, plug in the power supply. If everything is correct (SD card, Ethernet, power, ...) the LEDs should start blinking.

B- Web browser

If you connect correctly the RedPitaya, you should be able to load redpitaya home page in web browser. The web address is written on the redpitaya as shown in fig. 17 in below:

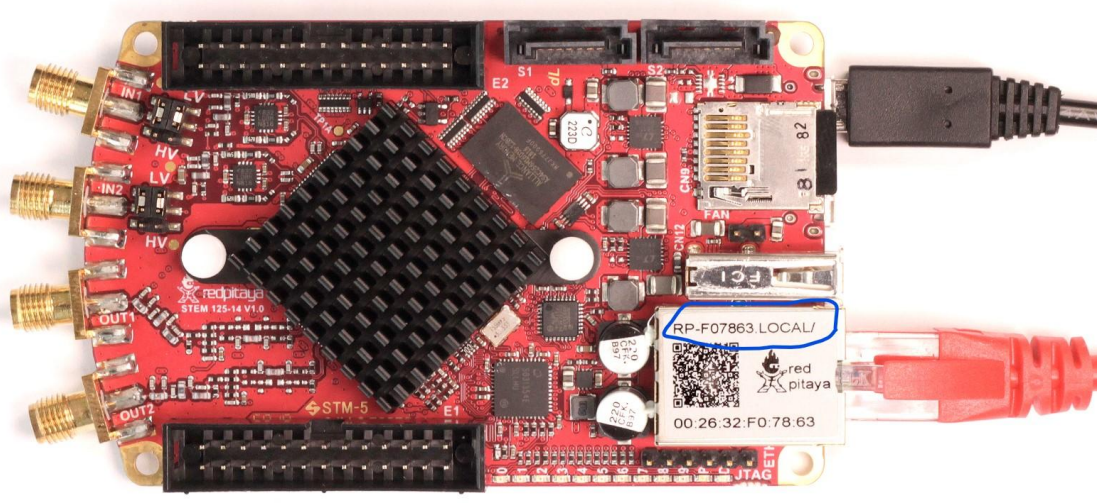

Fig. 17: Web address of RedPitaya to search in web browser.

When you search this address in the browser you should see the home page like fig. 18.

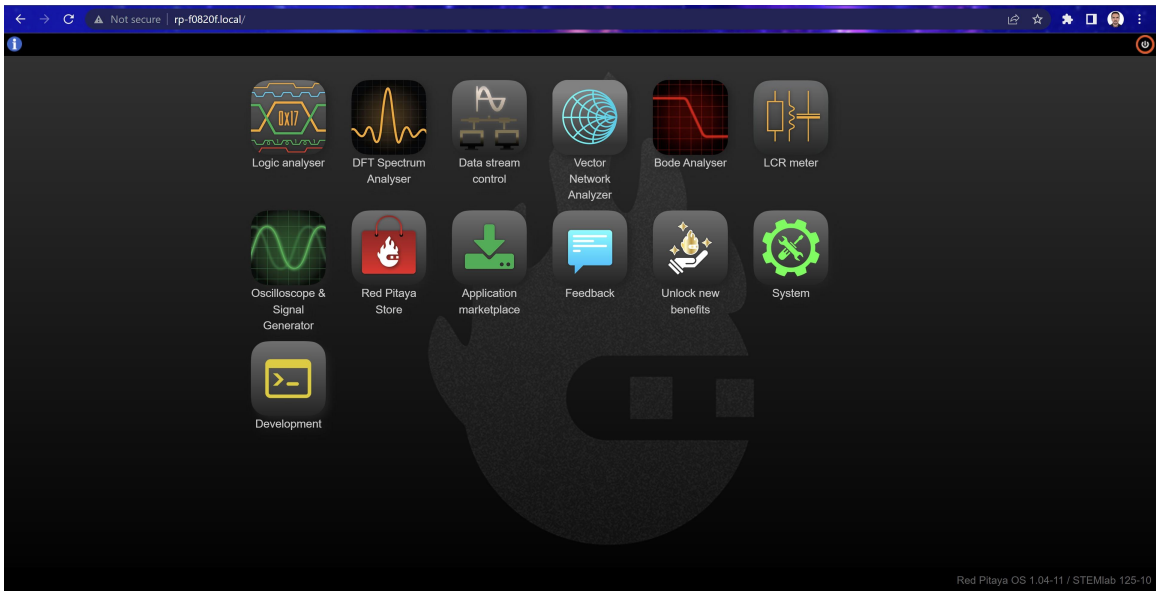

Fig. 18: Home page of redpitaya.

C- Finding IP address of redpitaya

For finding the IP address of redpitaya, one way is pinging to the host name or using arp -a command in the CMD but in this case you should know the MAC address of your redpitaya, fig. 19.

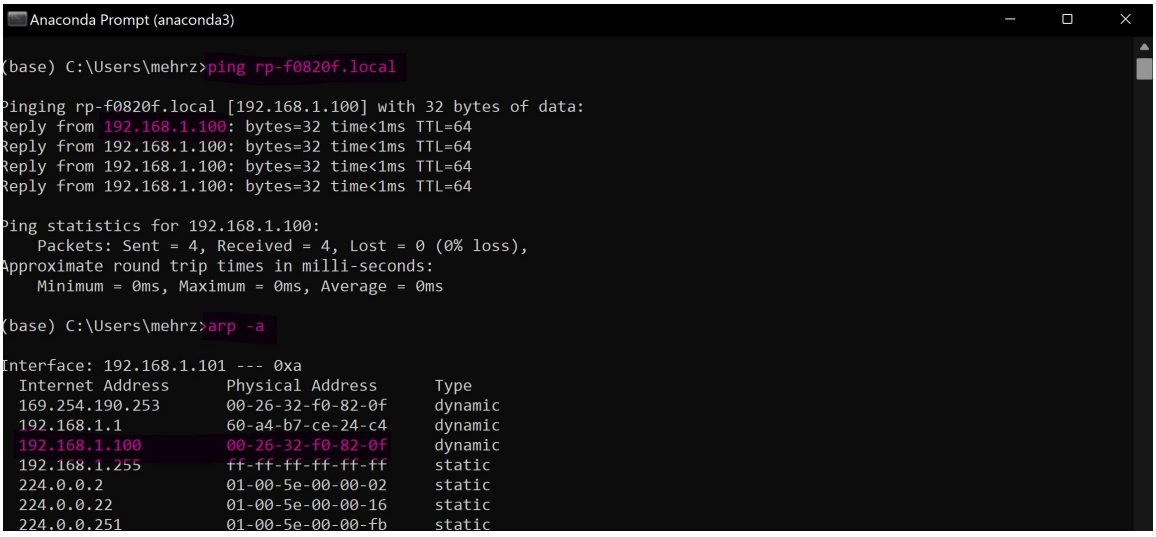

Fig. 19: Finding IP address of RedPitaya.

As it shown in the fig. 20, you can load home page by IP address.

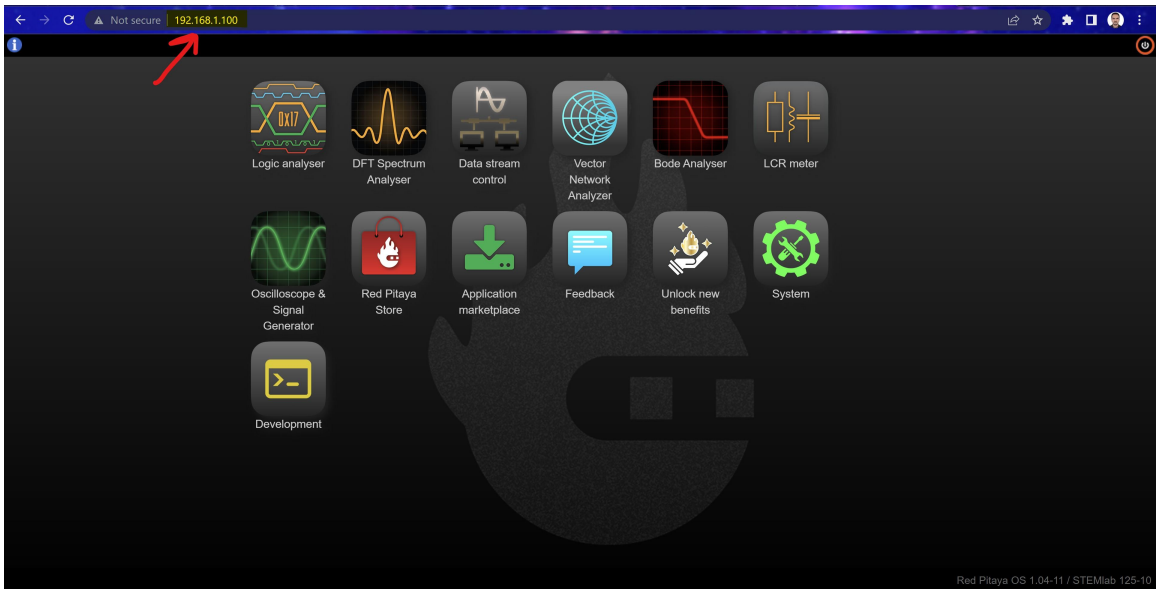

Fig. 20: Loading home page of RedPitaya by IP address.

D- Setting RedPitaya

Usually amplified signal of transmitted light after cavity is high voltage more than of 1 volt, therefore the jumper on the redpitaya board should be changed to high voltage (max 20 volt) and in this case signal have to be connected to input 1 and output 1 is also for scanning piezo voltage of the ECDL laser, fig. 21.

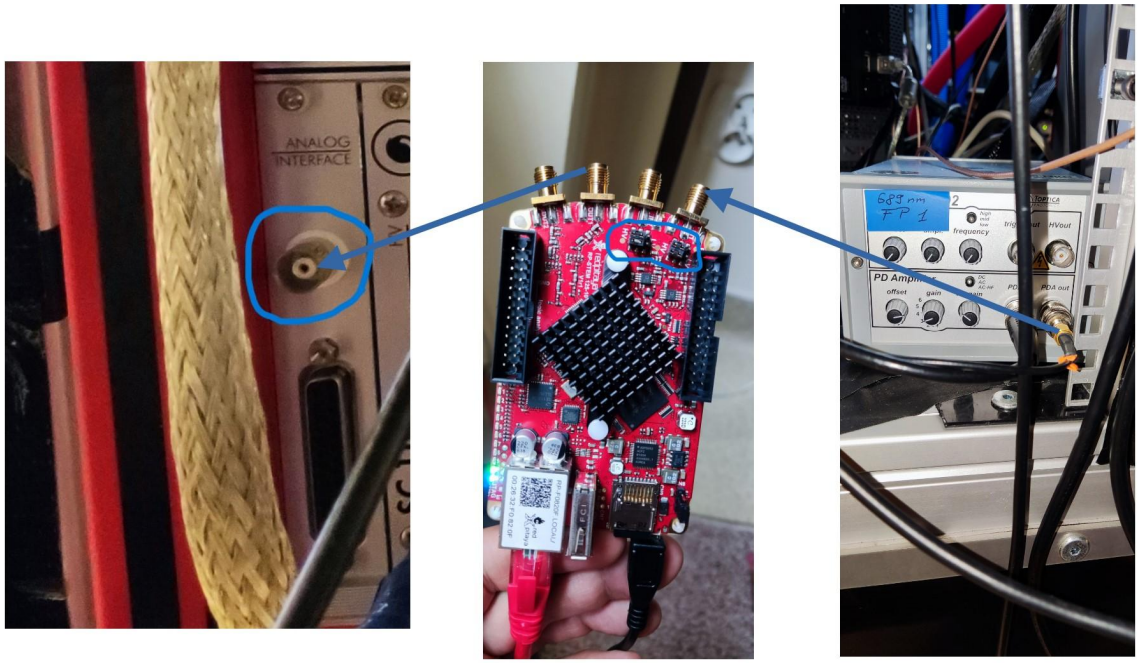

Fig. 21: Setting RedPitaya.

For changing piezo voltage I have connected output 1 on the redpitaya to an Analog Interface port of Toptica on the laser driver rack. To be sure that applied voltage on this interface is applied to the main piezo voltage of the scan controller, you need to check the jumper on the boards of Analog Interface

and Scan Controller before installing them on the rack. The jumper should be set as it recommended for changing the piezo voltage in their datasheet.

E- Installing Application

After preparing the hardware, now is the time to install the application on the RedPitaya system. For this case we have two option:

a- SD card

In this case you need to turn off the redpitaya and take out the SD card then insert it to your system by SD card reader. Then you can simply copy and paste your application folder to this directory of the SD card which is shown in the fig. 22 also:

www > apps

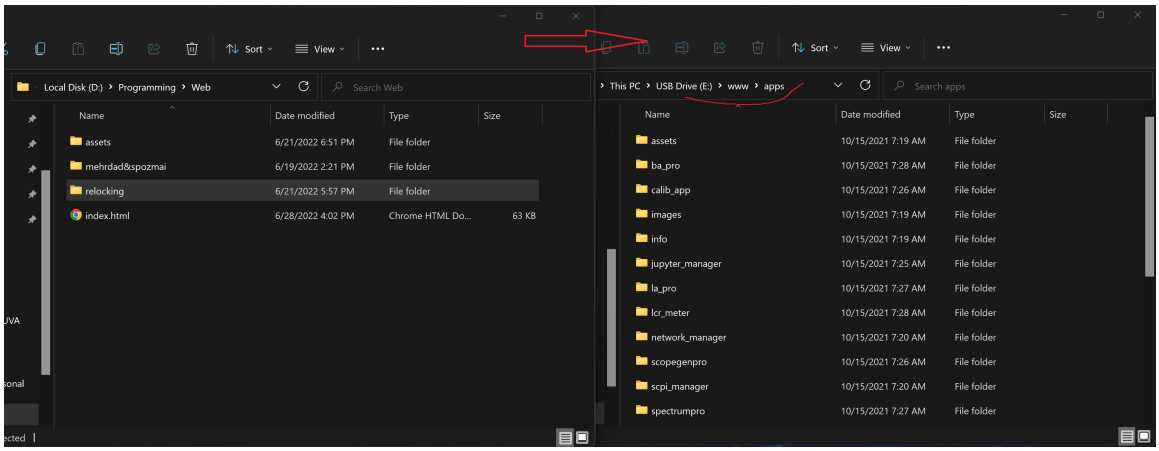

Fig. 22: Transfer application to the RedPitaya by SD card.

b- Putty and WinSCP on Windows or SSH connection on Linux

In this method you don't need to turn off the redpitaya, if you are working on Windows, install the Putty and WinSCP. For the putty and winscp use these configuration as shown in the fig. 23. Username and password to connect to the redpitaya is root and root, after successfully connection on the putty you have to execute this command to make file-system writable. Now you can upload your application to this directory which is shown in the fig. 24 also:

/opt/redpitaya/www/apps/

If you are working on Linux, execute this command to connect to your redpitaya at first:

ssh root@192.168.1.100

then use these command to transfer your application to the redpitaya:

cd /opt/redpitaya/www/apps
cp -r /home/programming/redpitaya/relocking ./relocking

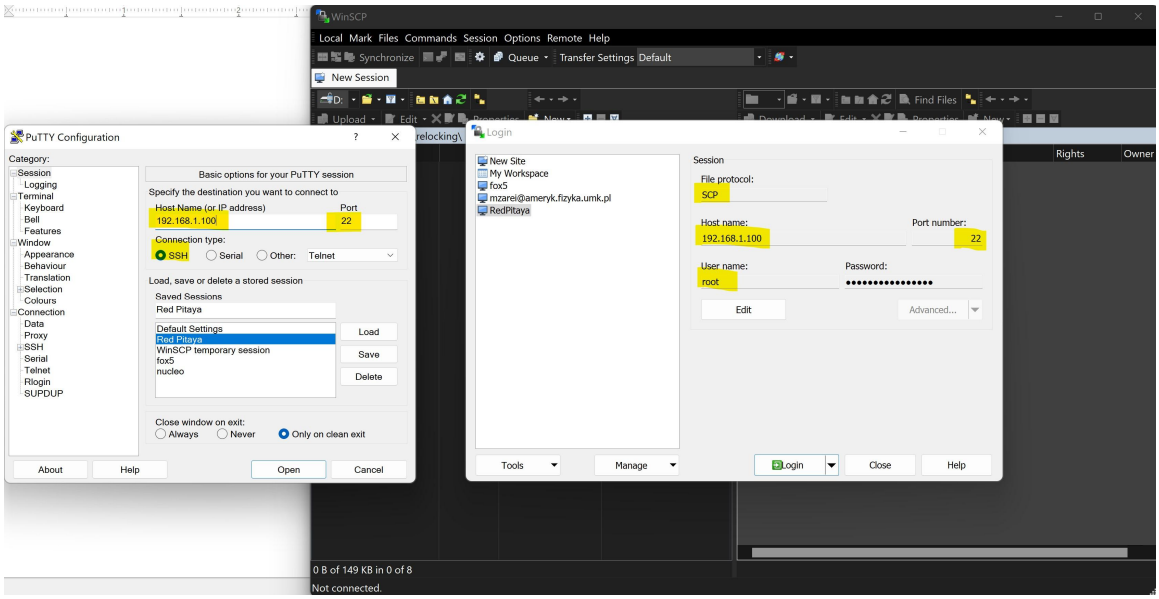

Fig. 23: Putty and WinSCP configuration to connect the redpitaya

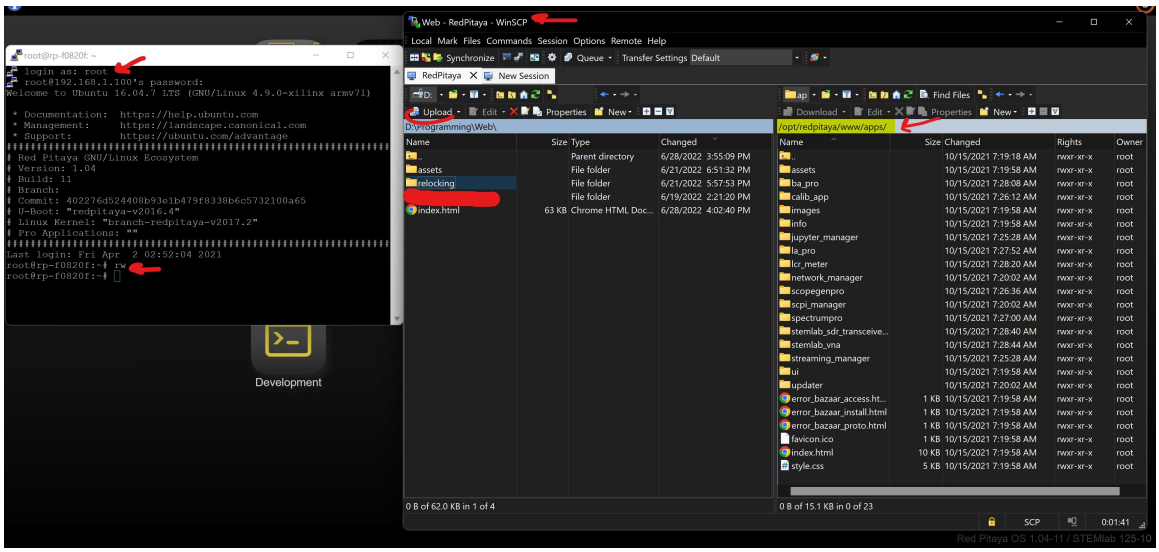

Fig. 24: Transferring application to the redpitaya through putty and winscp.

F- Compiling application

When you transfer the application to the redpitaya if you don't have the compiled file as shown in the fig. 25 or you changed something in the backend code, then you need to compile the code. For compiling the code you need to connect to the redpitaya board through putty on Windows or ssh connection in Linux then go to the directory of your application and execute this command like fig. 26:

make INSTALL_DIR=/opt/redpitaya

If everything is correct your application would be created, now you can go to the next step!
Note: check extra commands at the end for removing controllerhf.so.

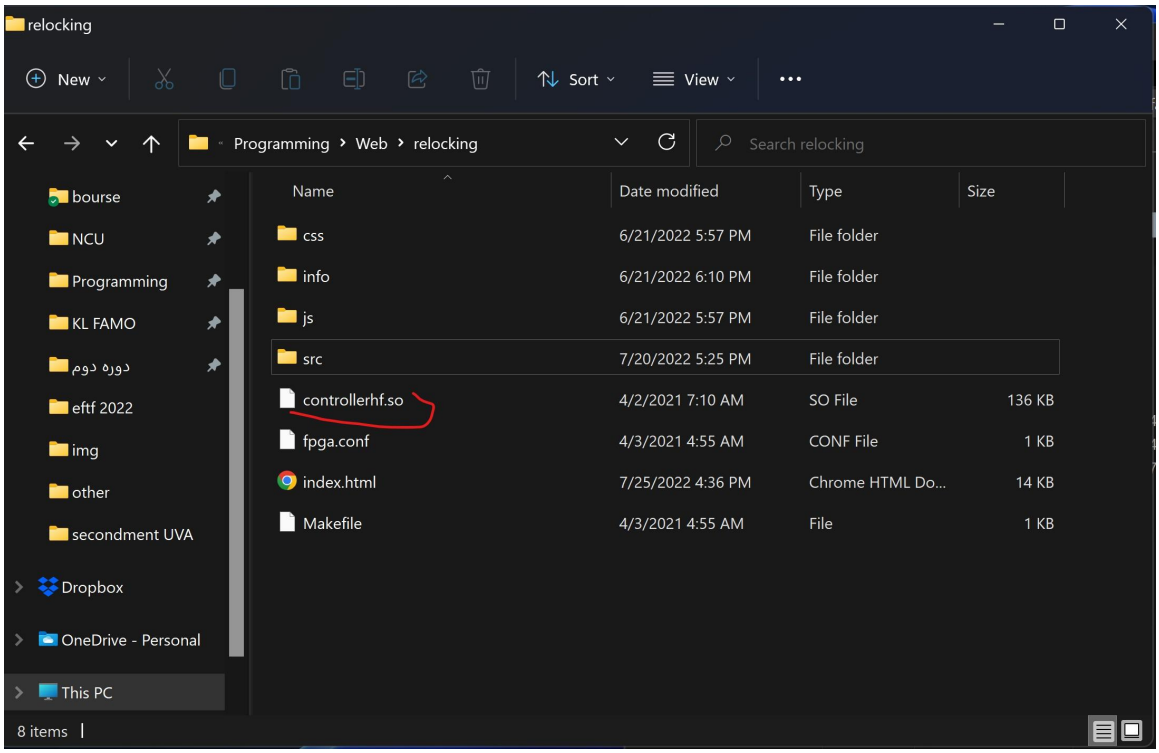

Fig. 25: Compiled application (controllerhf.so).

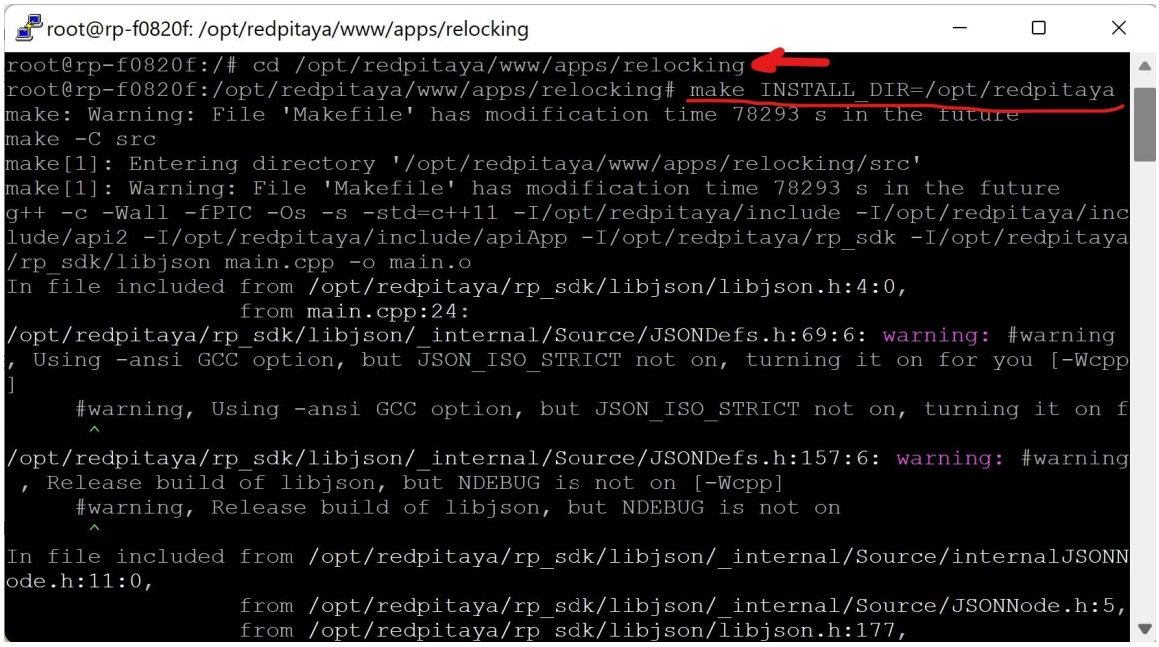

Fig. 26: Compiling the application.

G- Refresh RedPitaya

After refreshing the home page you should see the icon of your application in the home page fig. 27.

Fig. 27: Application in the home page.

H- Setting the application

Now you can load the web page related to the application. As is shown in fig. 28 you can change the mode of page to dark and in the left side there are some panels for plot setting, laser control, and wavemeter. In the plot setting we should click on channel 1 to get information of input 1 on the redpitaya (detected signal of cavity transmission), also gain level should be set on HV (high voltage), probe attenuation is depend to the cable you are using. Now if you click on the run button, application will start working. Laser control panel is for changing the piezo voltage, but if you are in the auto lock mode you are not able to change the piezo manually for this case you need to click on the manual lock first then would be able to change piezo manually and for auto lock you should click on the auto lock button.

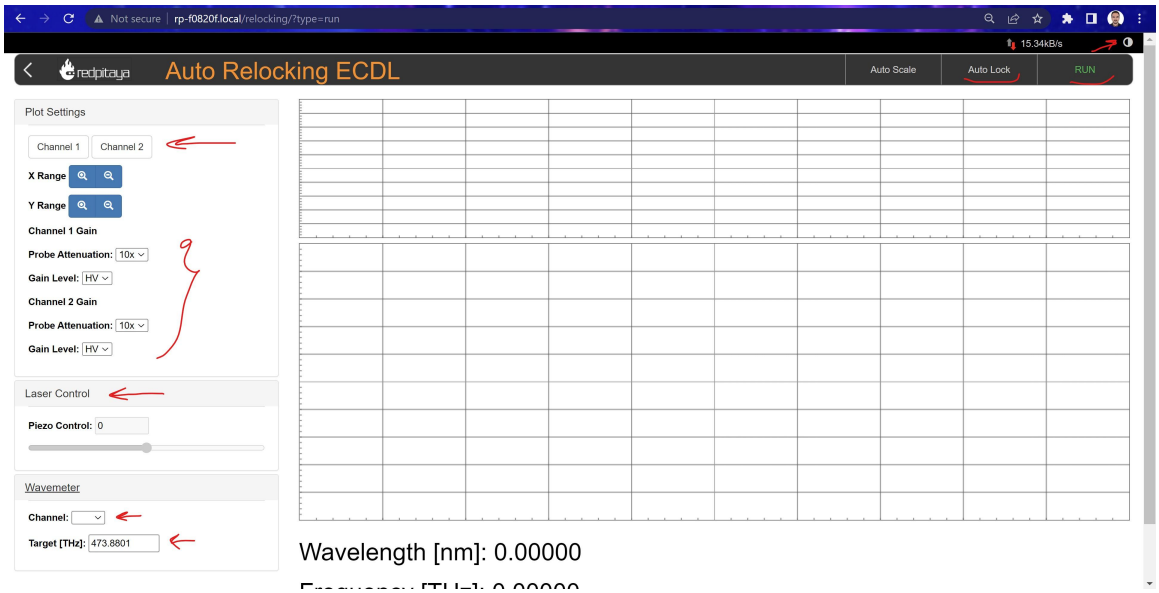

Fig. 28: Auto Relocking ECDL web page.

Note: RedPitaya outputs are limited to ±1 volt.

In the wavemeter panel you can select the channel which the laser light is connected and in the target you have to write your target frequency (application lock the laser base on this target).

I- Setting piezo control on 0

As outputs of redpitaya are limited to ±1 volt, we are not able to scan at least one single mode to another one, so to have a better feedback from relocking is recommended to set the piezo control on 0 at first in the application then lock the laser manually on the scan controller of laser driver and at the end click on the auto lock fig. 29.

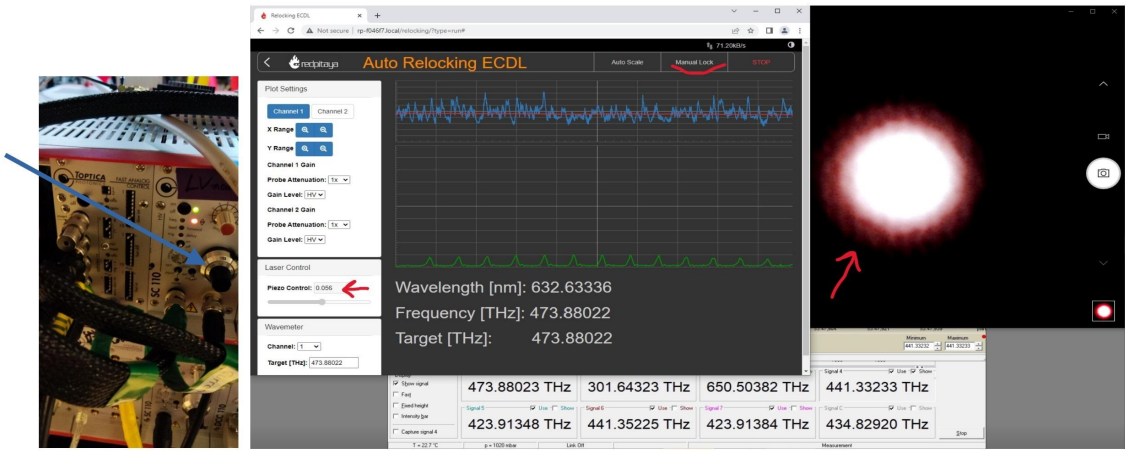

Fig. 29: Setting piezo control on 0.

Extra commands for part F:

Be sure that executed file (controllerhf.so) is removed then compile the new file, Fig 30.

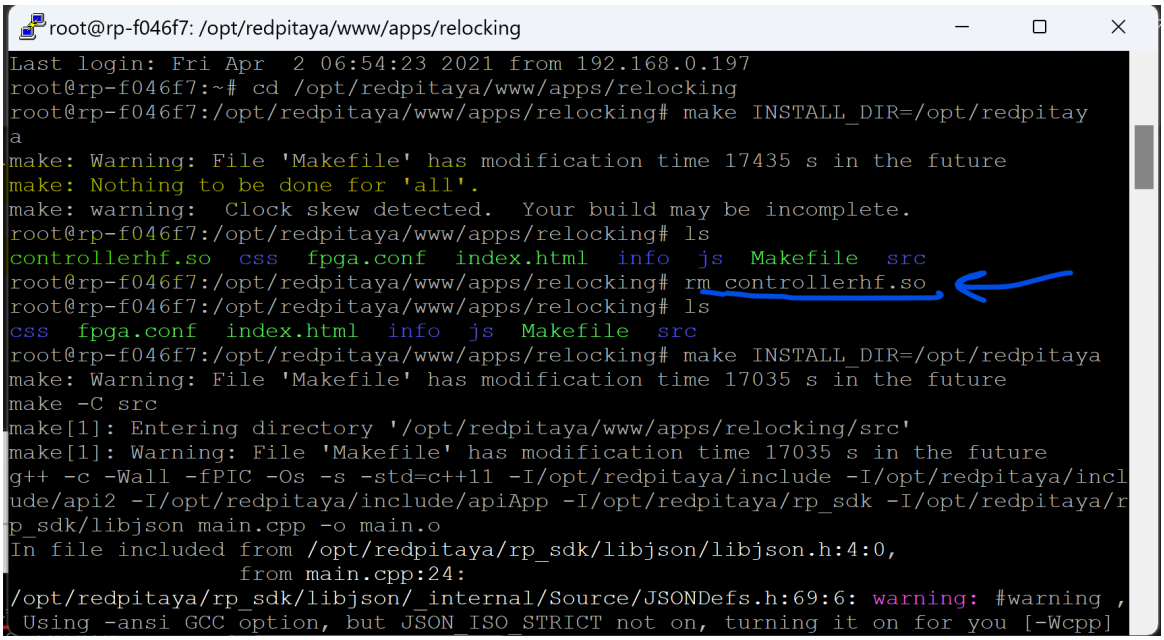

Fig. 30: Remove previous compiled file.