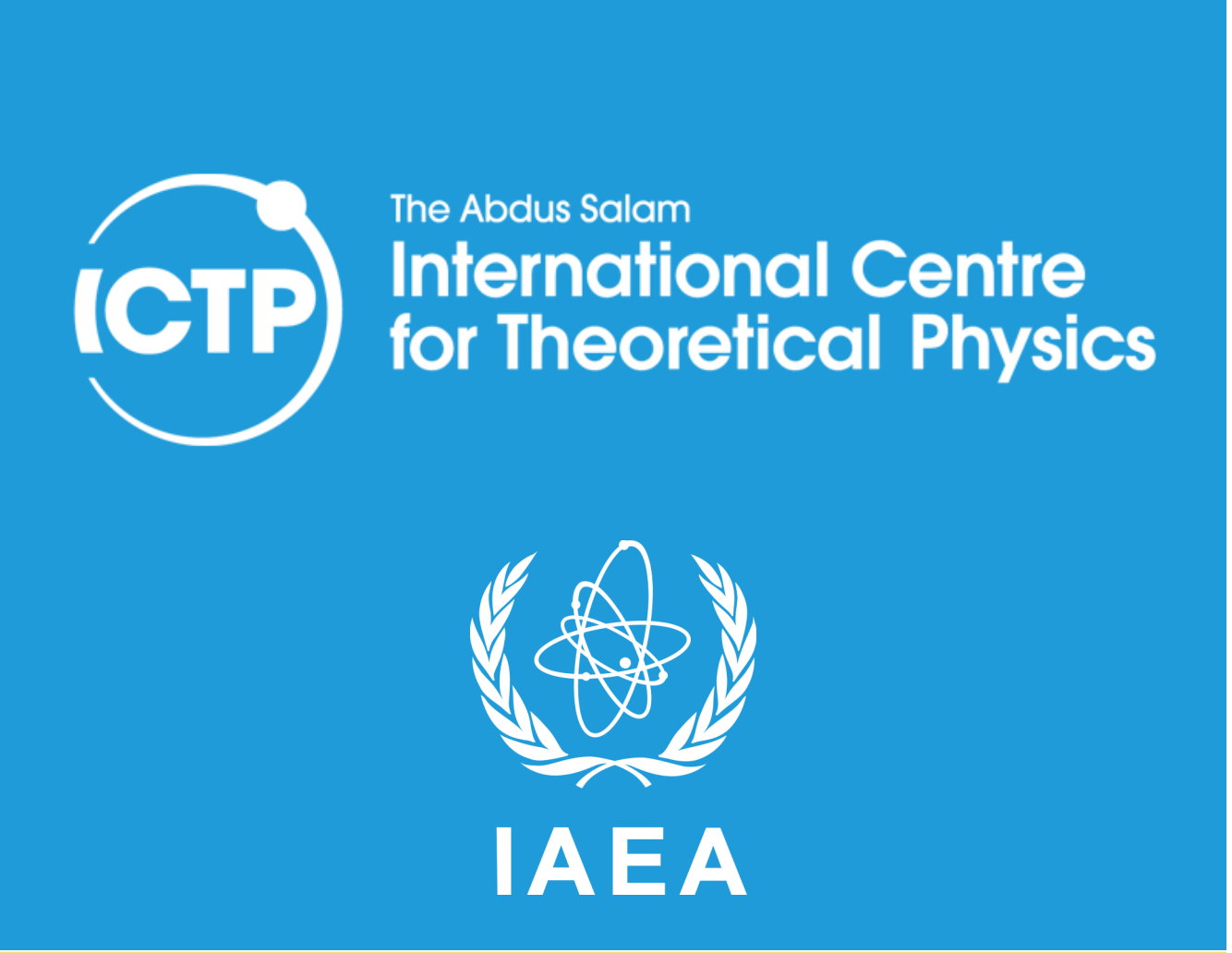


Joint ICTP-IAEA School on Systems-on-Chip based on FPGA for Scientific Instrumentation and Reconfigurable Computing | (smr 3891)

Before Starting (LAB 0)

Luis Guillermo García Ordóñez
Multidisciplinary Laboratory
ICTP

SMR3891 Tutors

Luis Garcia Ordoñez

Postdoctoral Fellow

+39 040 2240 9912 | lgarcia1@ictp.it

MLAB-202

Werner Florian

PhD Student

+39 040 2240 9913 | wflorian@ictp.it

MLAB-201

Maynor Ballina Escobar

TRIL fellow

+39-040-2240 9913 | mballina@ictp.it

MLAB-201

Romina Soledad Molina

Postdoctoral Fellows

+39 040 2240 9912 | rmolina@ictp.it

MLAB-202

Bruno Valinoti

PhD Student

+39 040 2240 9913 | bvalinot@ictp.it

MLAB-201

Iván René Morales Argueta

PhD Student

+39 040 2240 9912 | imorales@ictp.it

MLAB-202

Agustin Silva

PHD Student

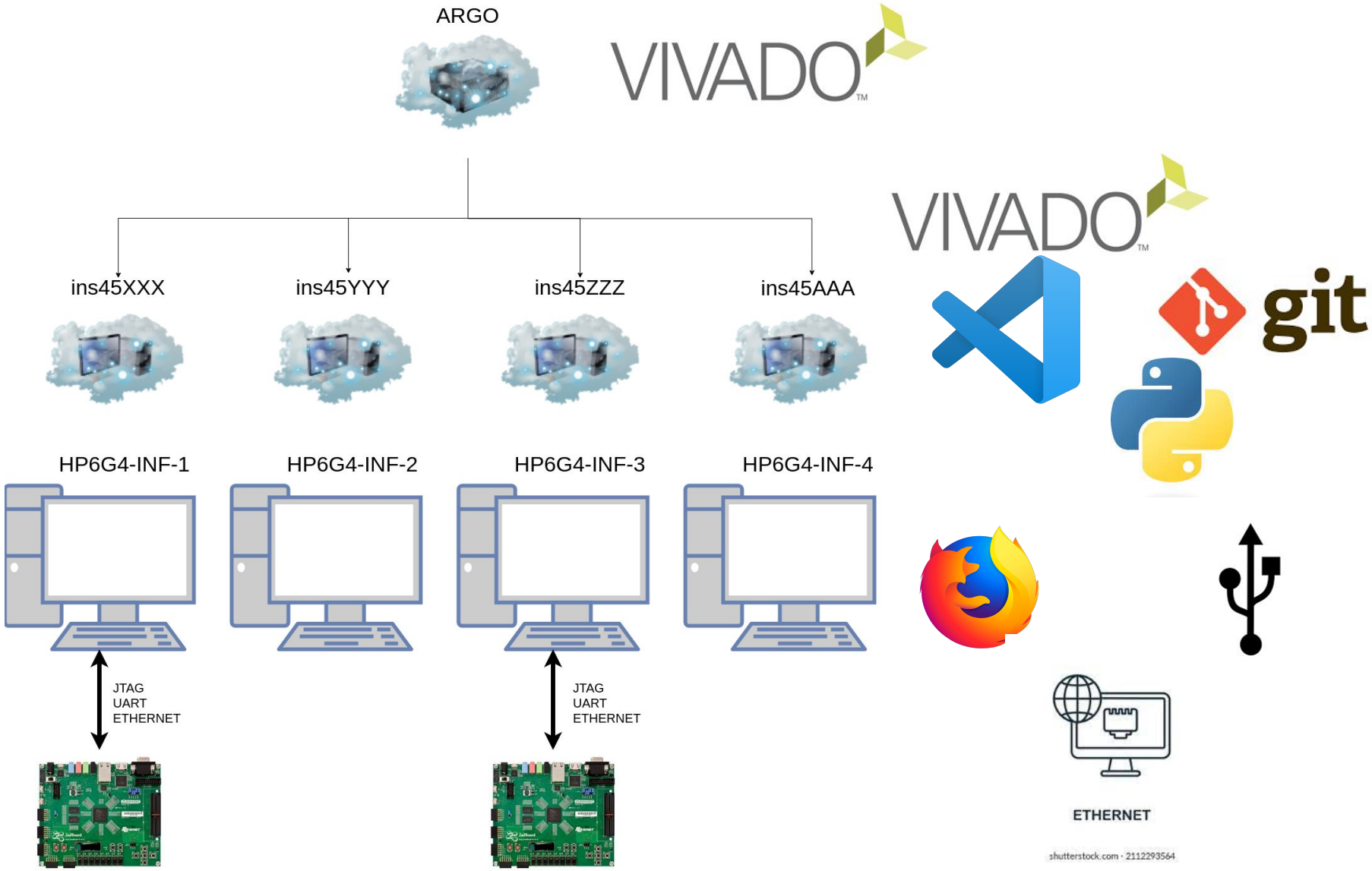
+39 040 2240 9902 | email@notavailableyet

MLAB-102

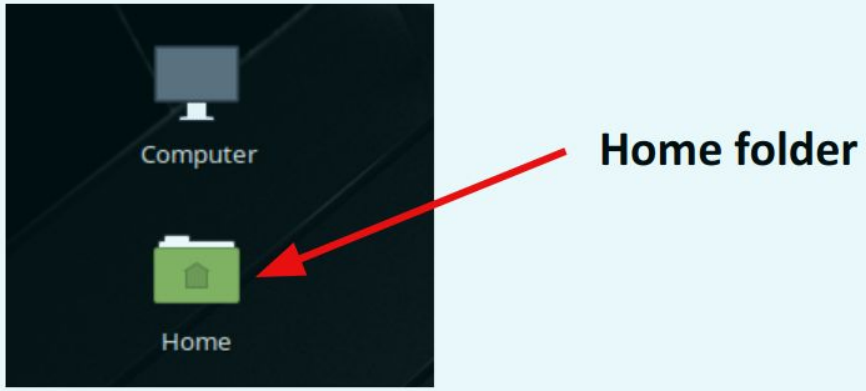
Our Lab Structure

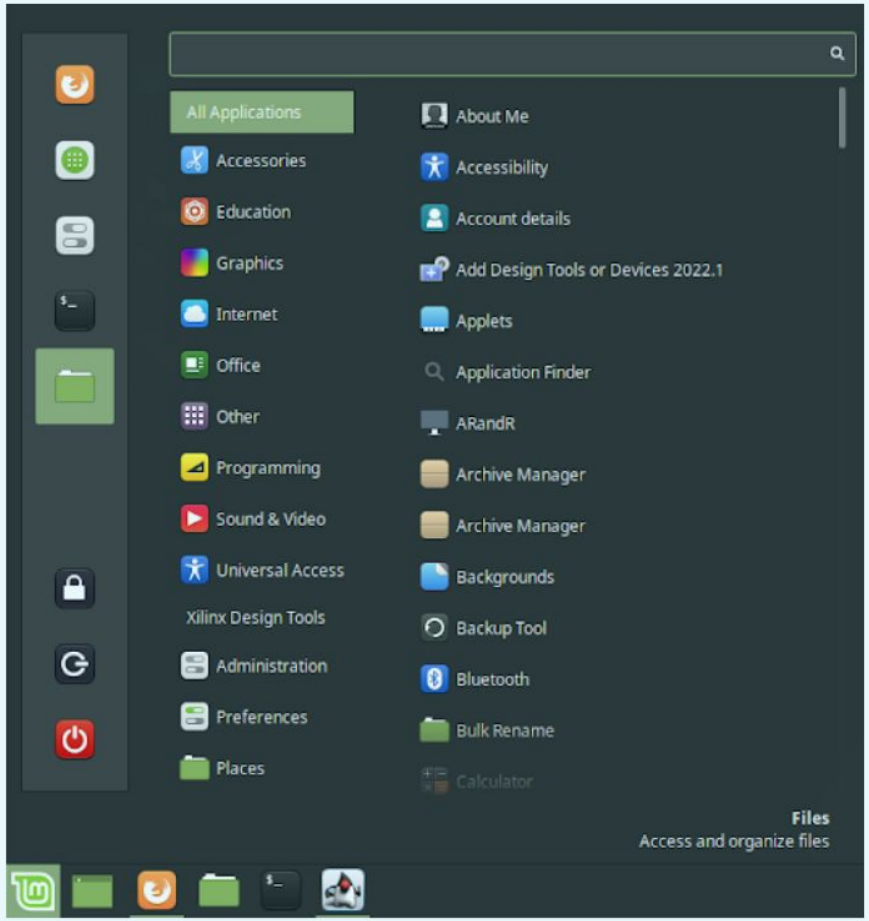
Virtual Machines

Local Machines



Knowing your Desktop (Cinnamon Environment)

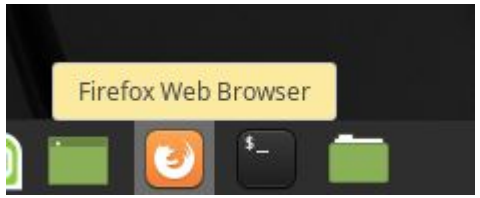


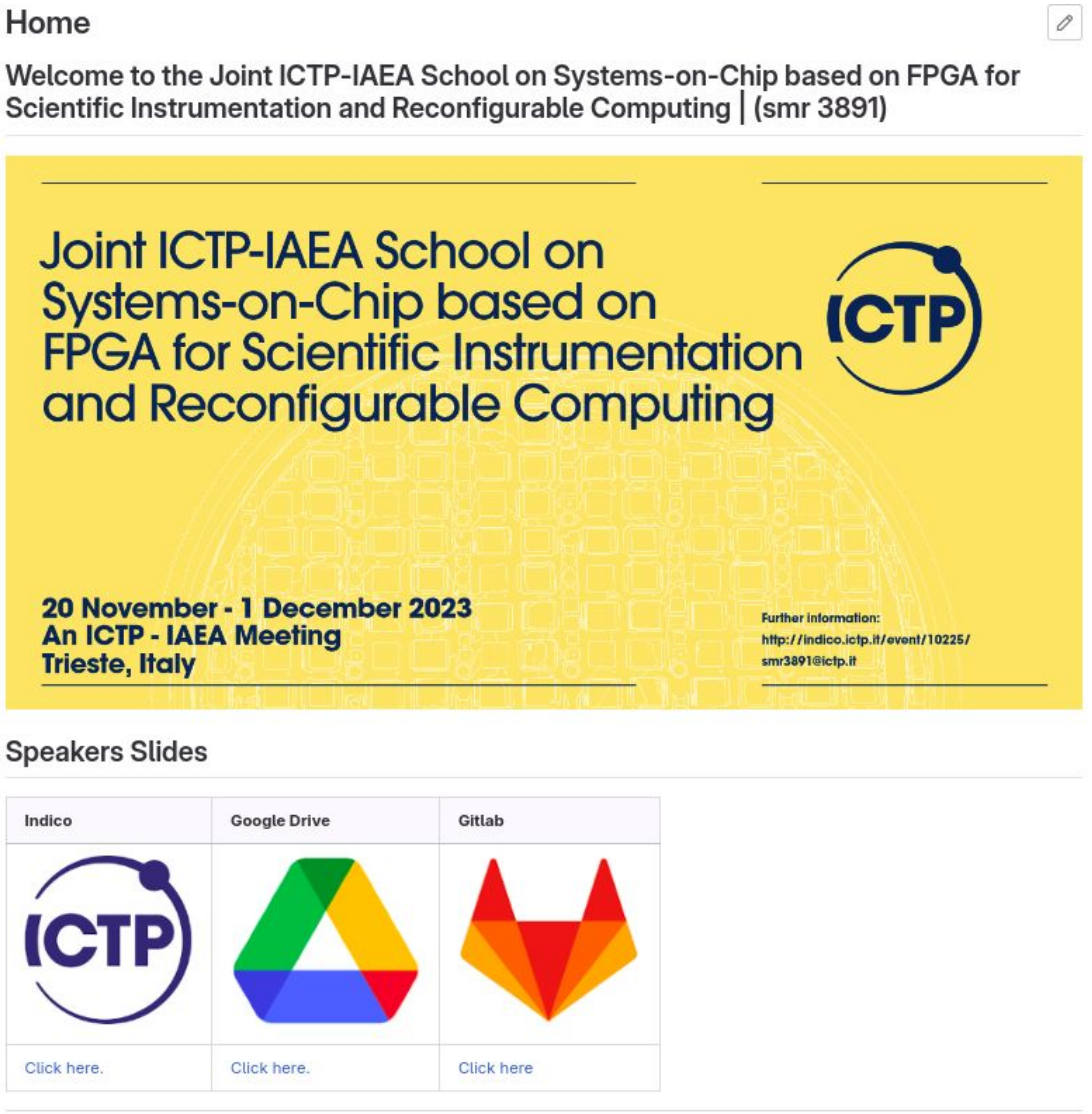


Checking the Wiki

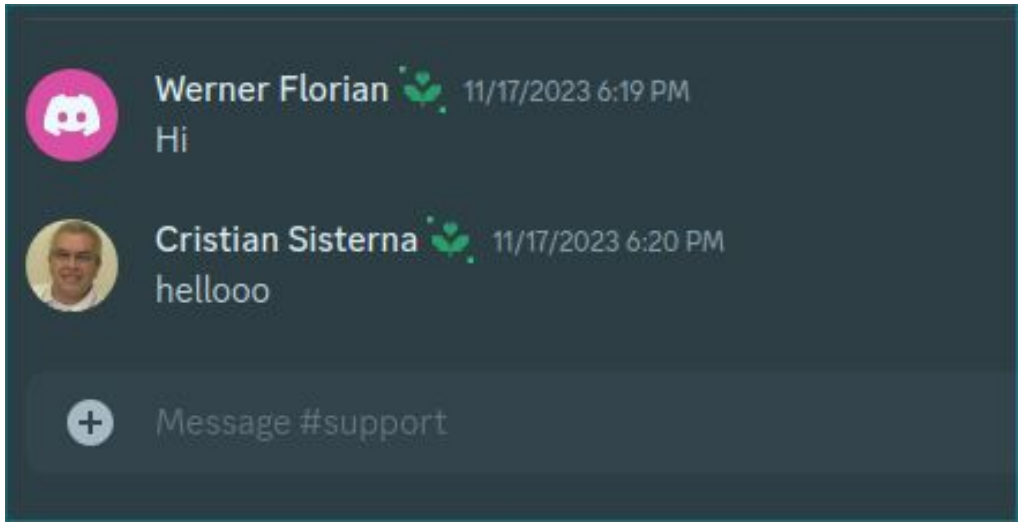
Gitlab Wiki
https://gitlab.com/ictp-mlab/smr-3891/-/wikis/home

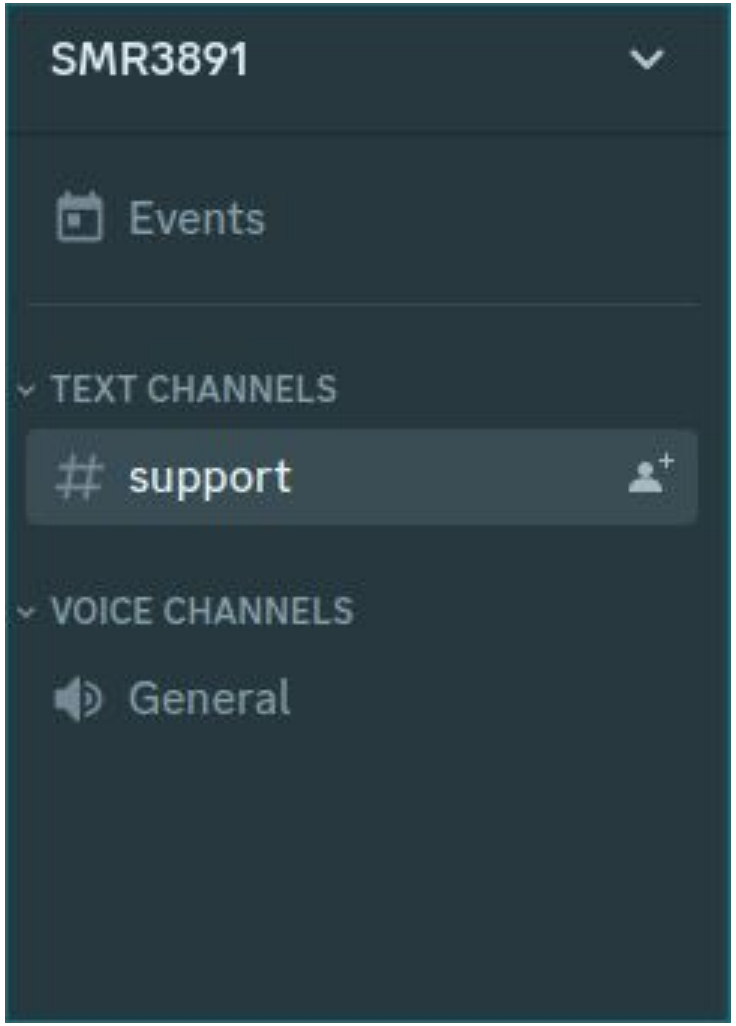
Gitlab Repository
https://gitlab.com/ictp-mlab/smr-3891





Discord communication Channel





https://discord.gg/3uXV7tTVE

Checking the Wiki

Before Starting · Wiki · IC

gitlab.com/ictp-mlab/smr-3891/-/wikis/Labs/Before-Starting

git - the simpl... Google Solved: HLS IP... 0 Notificaciones CERN Gmail YouTube Maps All Bookmarks

ICTP-MLAB › smr3891 › Wiki › ••• › Before Starting

Project
smr3891
Pinned
Issues 0
Merge requests 0
Wiki
Manage
Plan
Code
Build
Secure
Deploy
Operate
Monitor
Analyze
Settings
Help

Last edited by RomiSolMolina 1 day ago

Page history | New page

Before Starting

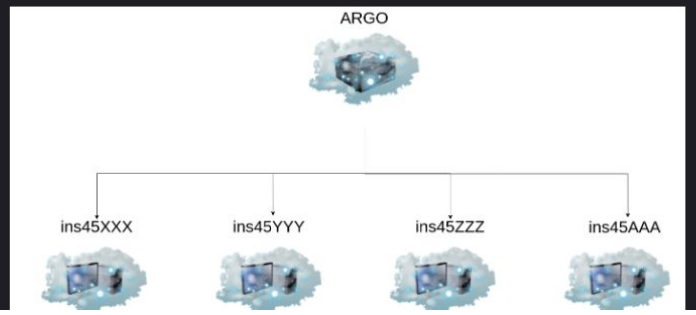
Joint ICTP-IAEA School on Systems-on-Chip based on FPGA for Scientific Instrumentation and Reconfigurable Computing

ICTP

Before Starting

Welcome to the Joint International Centre for Theoretical Physics (ICTP) and International Atomic Energy Agency (IAEA) School on Systems-on-Chip based on FPGA for Scientific Instrumentation and Reconfigurable Computing. We are thrilled to have you with us for these upcoming weeks!

The System-on-Chip/FPGA workflow often involves various tools written in different programming languages, requiring a specific work environment. To streamline this process, we've created a virtual machine that includes all the necessary tools and configurations for this workshop. The overal structure is shown in the figure bellow.



Clone repository
Edit sidebar

Home

Laboratory guides (revisar)

- Before Starting
- Lab 1: Hello World and GPIO IP Cores
- Lab 2: Communication Block (ComBlock) and RTL instantiation
- Lab 3: SoC-FPGA Development Framework: UDMA & Jupyter Notebook
- Lab 4: SoC-FPGA DAQ system and TCL
- Project Lab 1: Digital Pulse Processing for Isotope Identification
  - Stage 1.1: Pulse acquisition and detector characterization
  - Stage 1.2: Digital pulse processing and isotope identification
  - Stage 1.3: Baseline restorer and pile-up rejection
- Project Lab 2: Digital Pulse Processing for X-ray Photon Detection and Energy Measurement
  - Digital Pulse Processor (DPP)

Virtual environment

Additionally to your ICTP credentials you have been given a *username* and *password* for your virtual machine.

The typical format is the following:

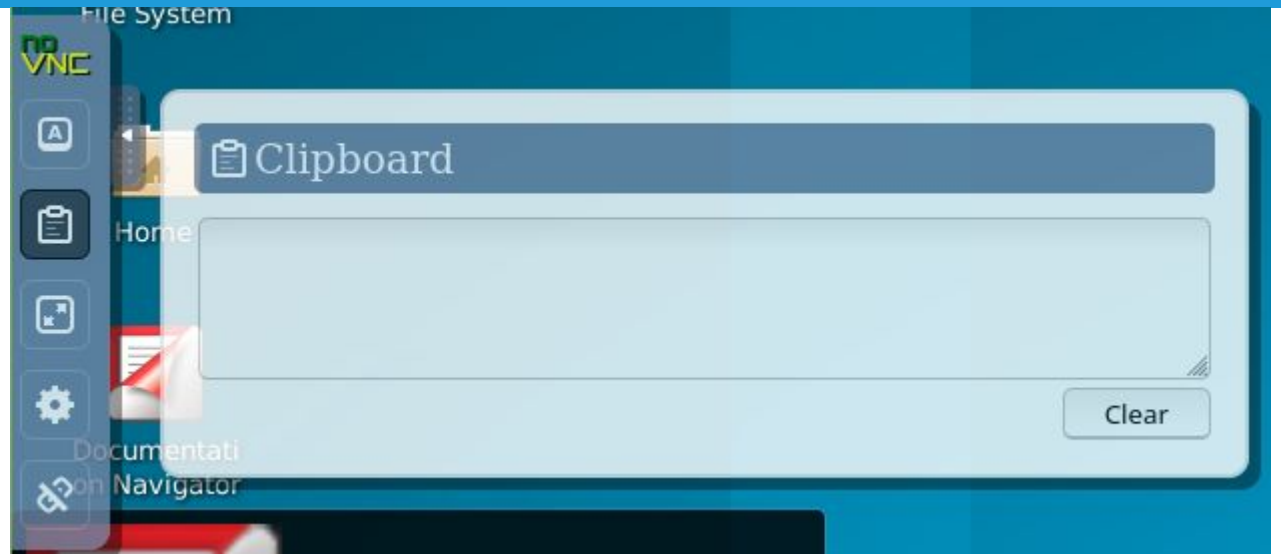
```
user: insXXXXX
password: xxxxxx
```

To enter into the virtual machine you should type in the explorer bar the following url:

```
https://insXXXXX.ictp.it
```

In this **example** the user is `ins45234`, and the input in the explorer bar should look like this:


noVNC — Mozilla Firefox
noVNC
https://ins45150.ictp.it
The Abdus Salam
International Centre
for Theoretical Physics
SMR 3891
Connect


File System
noVNC
Clipboard
Home
Clear
Documentati
on Navigator


File System
noVNC
Settings
Shared Mode
View Only
Clip to Window
Scaling Mode:
Remote Resizing
Advanced
Home
Documentati
on Navigator

The Git repository and the Virtual Machine.

Once we got access into the VM, let's proceed to create an environment suited for the school.

As mentioned before all the material for the laboratories will be handle in a **git** repository. Let's start by **cloning** the repository into our VMs.

1. Let's create and enter into a new directory called `smr3891` in our **Home** folder. There are two ways of doing this:

- **Option 1:** By opening a `Terminal` in `Applications->Terminal Emulator` and running the following command:

```
mkdir ~/smr3891
cd ~/smr3891
```

- **Option 2:** By using the graphic interface, open the **Home** directory in your desktop, right click, and selecting the option 

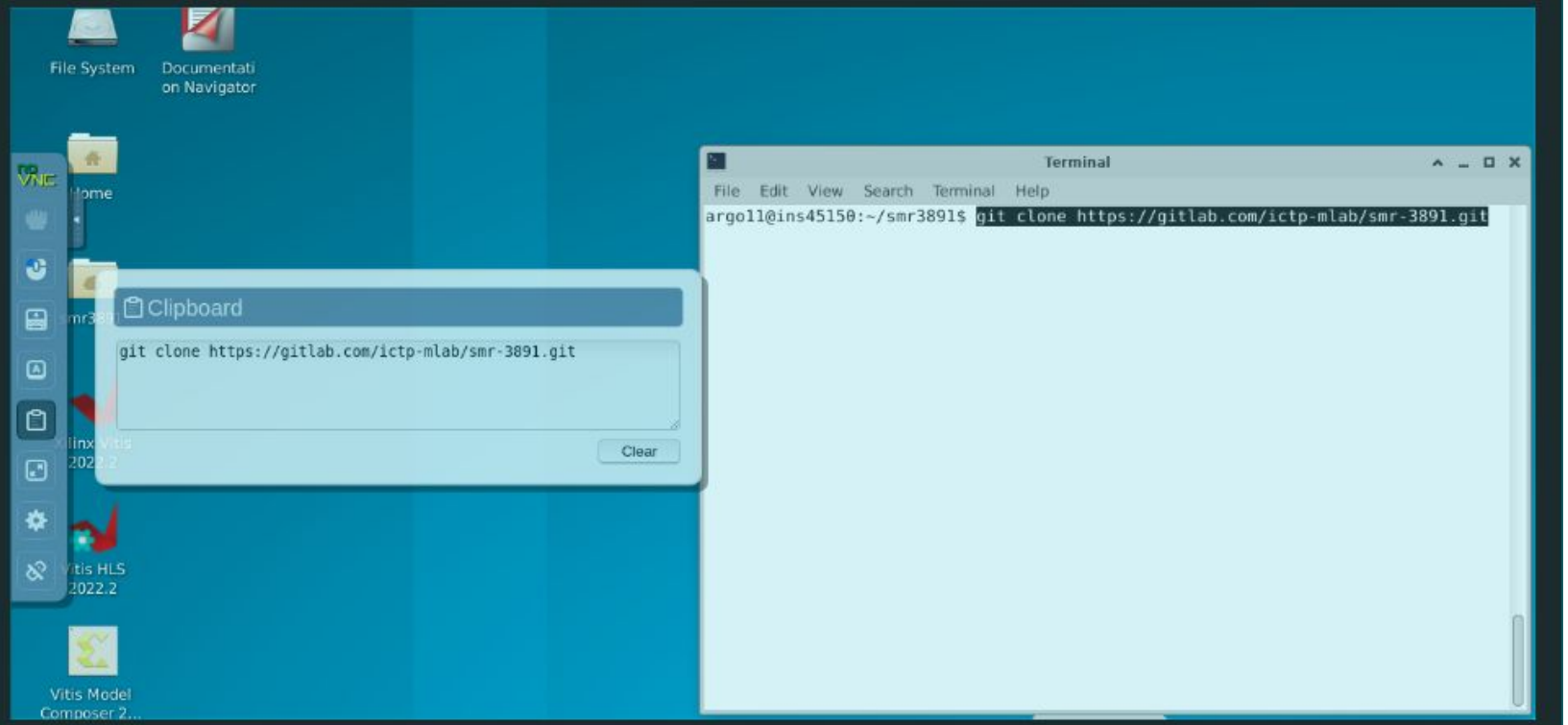
. Then, name it as `smr3891`.

Cloning the Repository

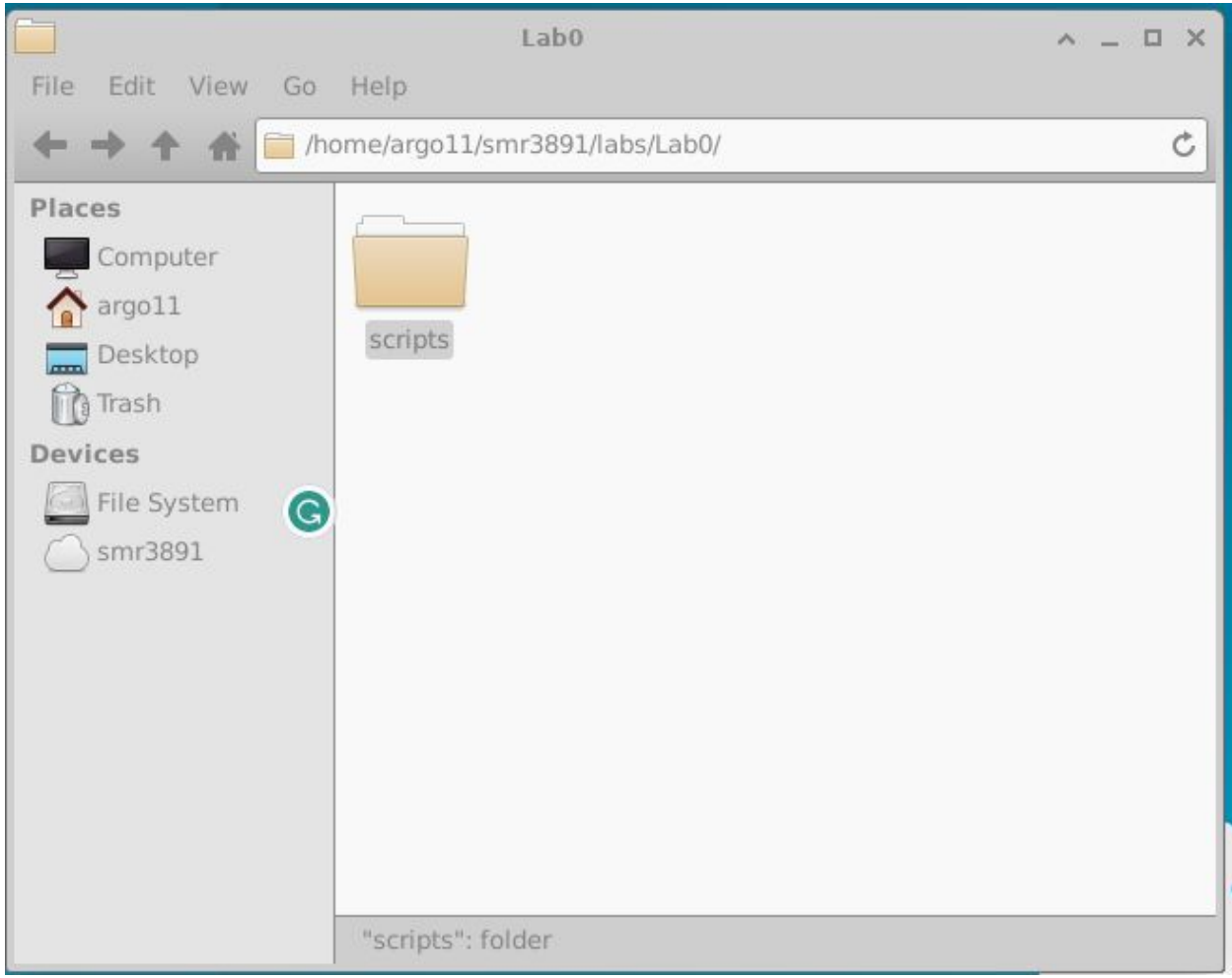
3. Copy the following code into the VM clipboard:

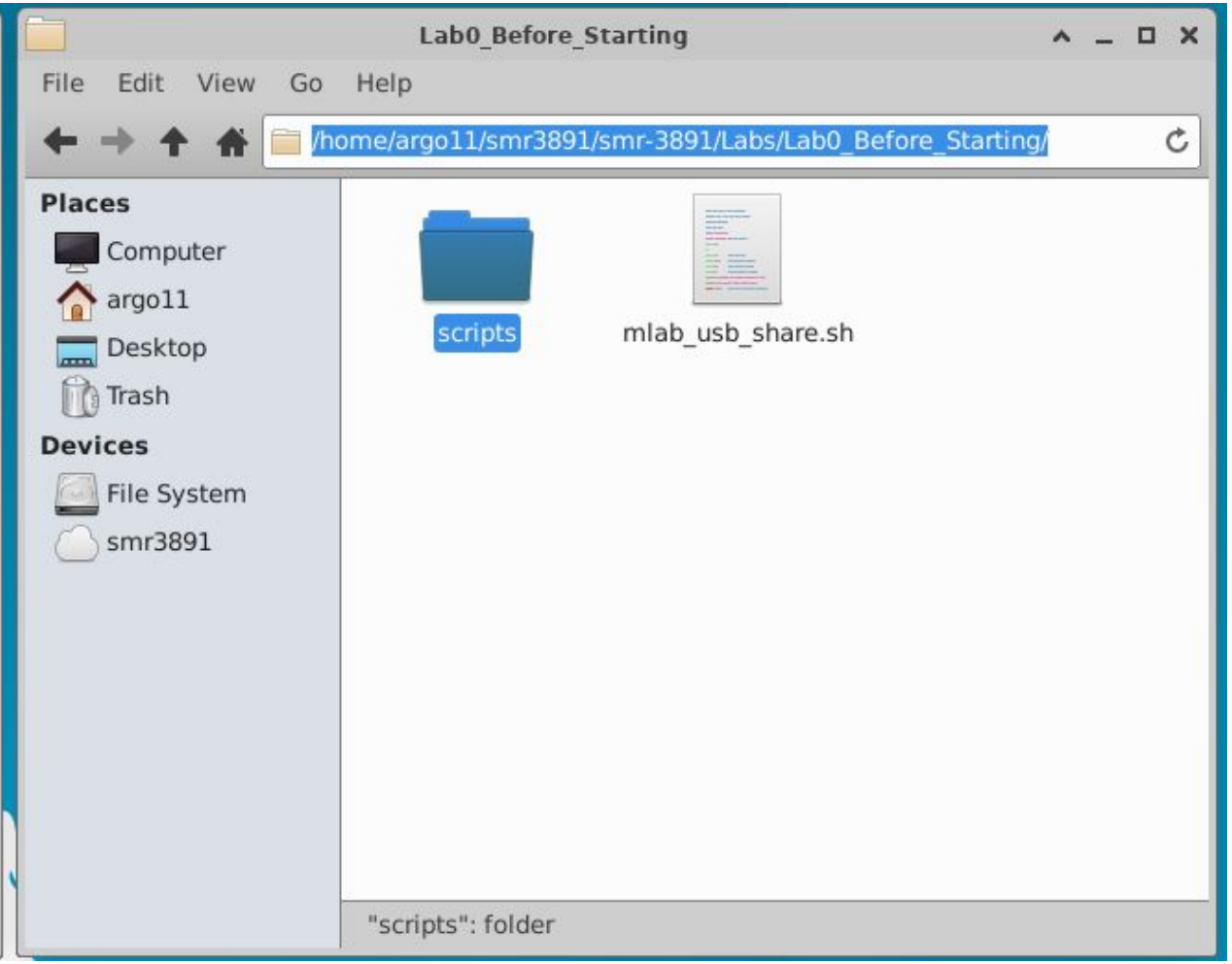
```
git clone https://gitlab.com/ictp-mlab/smr-3891.git
```

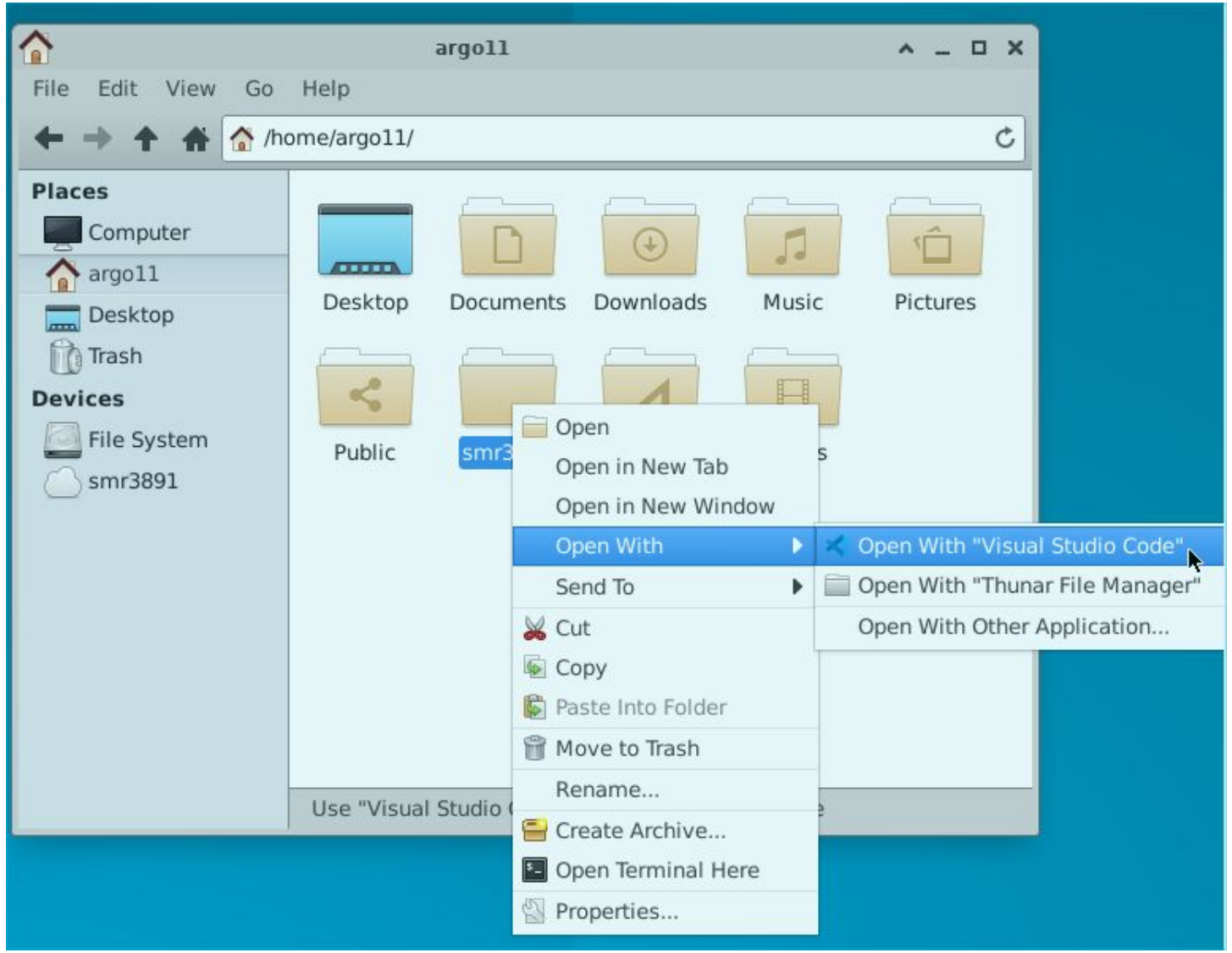
and paste it into the open terminal. ****PRO-TIP:** paste into the terminal using `Ctrl+Shift+v`.



```
/home/<argo_user>/
└── smr3891/
    ├── labs/
    │   ├── Lab0/
    │   │   └── scripts/
    │   │       └── Lab0_Before_Starting.ipynb
    │   ├── Lab1/
    │   │   ├── <Vivado_prj_name>
    │   │   ├── ...
    │   │   └── ...
    │   └── Lab2/
    │       └── ...
    ├── smr-3891 / #Git Repository
    │   ├── Labs
    │   ├── Readme.md
    │   └── ...
    ├── <git_repo_1>
    ├── <git_repo_2>
    └── ...
```

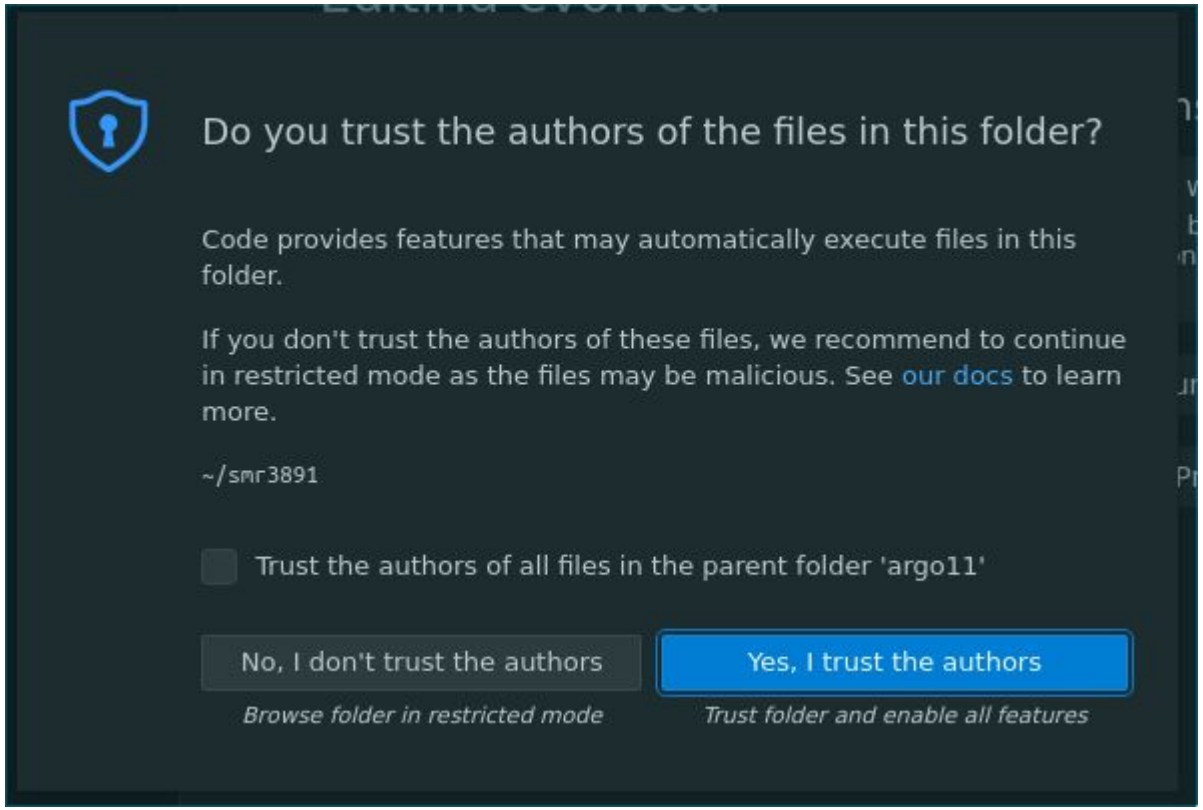

Lab0
File Edit View Go Help
/home/argo11/smr3891/labs/Lab0/
Places
Computer
argo11
Desktop
Trash
Devices
File System
smr3891
scripts
"scripts": folder


Lab0_Before_Starting
File Edit View Go Help
/home/argo11/smr3891/smr-3891/Labs/Lab0_Before_Starting/
Places
Computer
argo11
Desktop
Trash
Devices
File System
smr3891
scripts
mlab_usb_share.sh
"scripts": folder

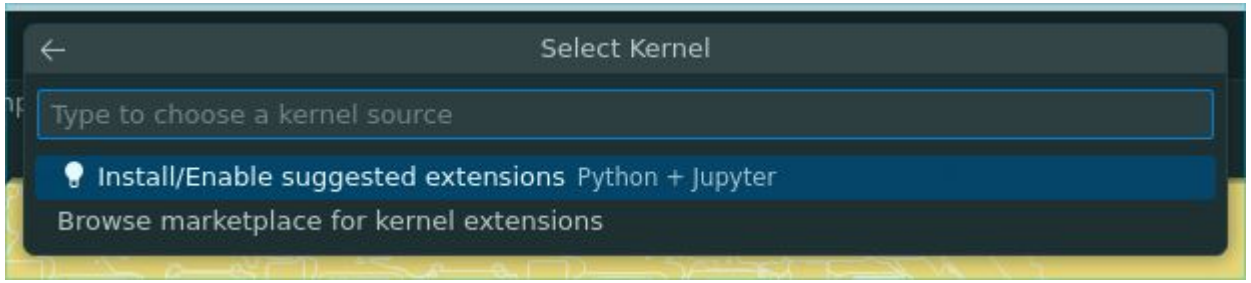

argo11
File Edit View Go Help
/home/argo11/
Places
Computer
argo11
Desktop
Trash
Devices
File System
smr3891
Desktop
Documents
Downloads
Music
Pictures
Public
Open
Open in New Tab
Open in New Window
Open With
Send To
Cut
Copy
Paste Into Folder
Move to Trash
Rename...
Create Archive...
Open Terminal Here
Properties...
Open With "Visual Studio Code"
Open With "Thunar File Manager"
Open With Other Application...

Visual Studio Code First Use

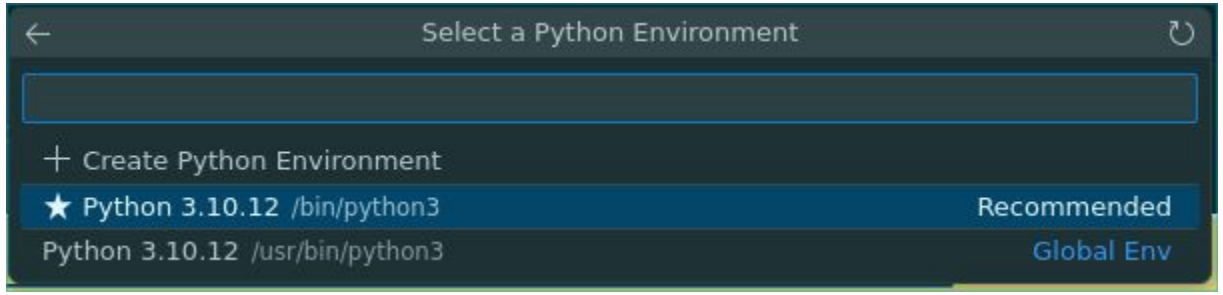
1. Trust the Sources

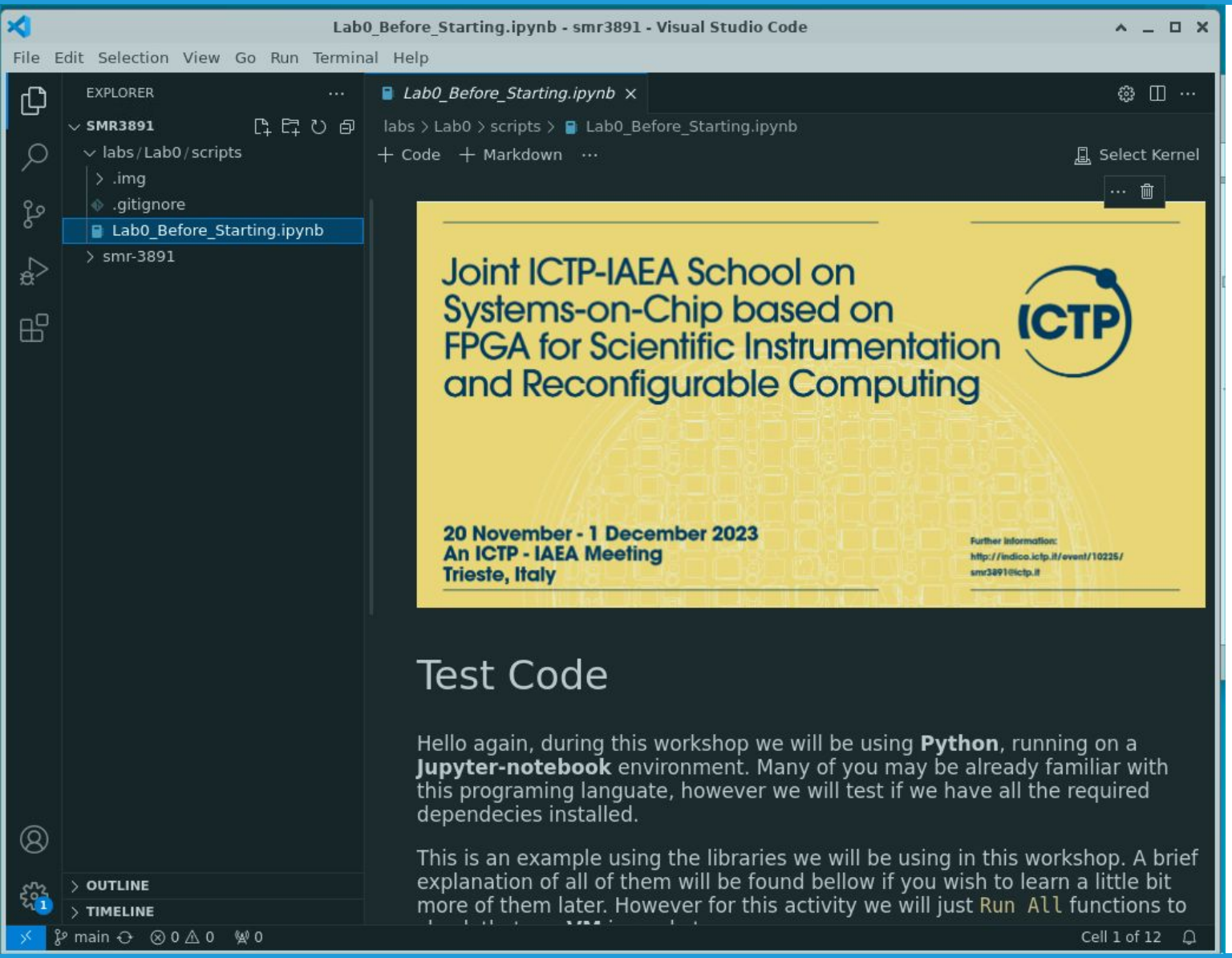


2. Install Extensions and Restart VSCode



3. Select Python Kernel




Lab0_Before_Starting.ipynb - smr3891 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
SMR3891
labs / Lab0 / scripts
.img
.gitignore
Lab0_Before_Starting.ipynb
smr-3891
Lab0_Before_Starting.ipynb
labs > Lab0 > scripts > Lab0_Before_Starting.ipynb
+ Code + Markdown
Select Kernel
Joint ICTP-IAEA School on Systems-on-Chip based on FPGA for Scientific Instrumentation and Reconfigurable Computing
ICTP
20 November - 1 December 2023
An ICTP - IAEA Meeting
Trieste, Italy
Further information:
http://indico.ictp.it/event/10225/
smr3891@ictp.it
Test Code
Hello again, during this workshop we will be using Python, running on a Jupyter-notebook environment. Many of you may be already familiar with this programing languate, however we will test if we have all the required dependecies installed.
This is an example using the libraries we will be using in this workshop. A brief explanation of all of them will be found bellow if you wish to learn a little bit more of them later. However for this activity we will just Run All functions to
OUTLINE
TIMELINE
main
Cell 1 of 12

Run All

Connecting to The Zedboard (local PC)

The Zedboard JTAG programmer and UART port are the following devices:

```
Bus 003 Device 010: ID 0403:6014 Future Technology Devices
International, Ltd FT232H Single HS USB-UART/FIFO IC

Bus 003 Device 008: ID 04b4:0008 Cypress Semiconductor Corp.
```

For network connectivity, we will be using an ethernet dongle that can be **ONE** of these two devices shown below:

```
Bus 004 Device 003: ID 0b95:1790 ASIX Electronics Corp. AX88179 Gigabit
Ethernet

Bus 002 Device 011: ID 0bda:8153 Realtek Semiconductor Corp. RTL8153
Gigabit Ethernet Adapter
```

Connecting to The Zedboard (local PC)

The Zedboard JTAG programmer and UART port are the following devices:

```
Bus 003 Device 010: ID 0403:6014 Future Technology Devices
International, Ltd FT232H Single HS USB-UART/FIFO IC

Bus 003 Device 008: ID 04b4:0008 Cypress Semiconductor Corp.
```

For network connectivity, we will be using an ethernet dongle that can be **ONE** of these two devices shown below:

```
Bus 004 Device 003: ID 0b95:1790 ASIX Electronics Corp. AX88179 Gigabit
Ethernet

Bus 002 Device 011: ID 0bda:8153 Realtek Semiconductor Corp. RTL8153
Gigabit Ethernet Adapter
```

Connecting to The Zedboard (local PC)

```
ictp_share_usb_device share <device_id_0>
<device_id_1> ...
```

```
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 003: ID 0b95:1790 ASIX Electronics Corp. AX88179 Gigabit Ethernet
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 0403:6014 Future Technology Devices International, Ltd FT
232H Single HS USB-UART/FIFO IC
Bus 003 Device 003: ID 04b4:0008 Cypress Semiconductor Corp.
Bus 003 Device 002: ID 0c45:6340 Microdia Camera
Bus 003 Device 008: ID 046d:c05b Logitech, Inc. M-U0004 810-001317 [B110 Optical
 USB Mouse]
Bus 003 Device 007: ID 03f0:354a Hewlett-Packard
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

```
ictp_share_usb_device share 0403:6014 04b4:0008
0b95:1790
```

Connecting to The Zedboard (local PC)

Did you get an error?

```
$ sudo apt update
$ ictp_share_usb_device share 0403:6014 04b4:0008
0b95:1790
```

Connecting to The Zedboard (local PC)

Too complicated? Let's go to the Wiki:

1. Open a terminal and run the following lines:

```
wget https://dbox.ictp.it/index.php/s/usb_share/download/mlab_usb_share.sh
chmod +x mlab_usb_share.sh
```

2. Now you can mount the zedboard ports and the USB Ethernet dongle using the following command:

```
./mlab_usb_share.sh
```

Connecting to The Zedboard (local PC)

Don’t worry, this configuration has to be done only once, unless you shut down the Zedboard or restart the PC.

So please **Do not Shut Down the Zedboards, or Disconnect them from their PCs**

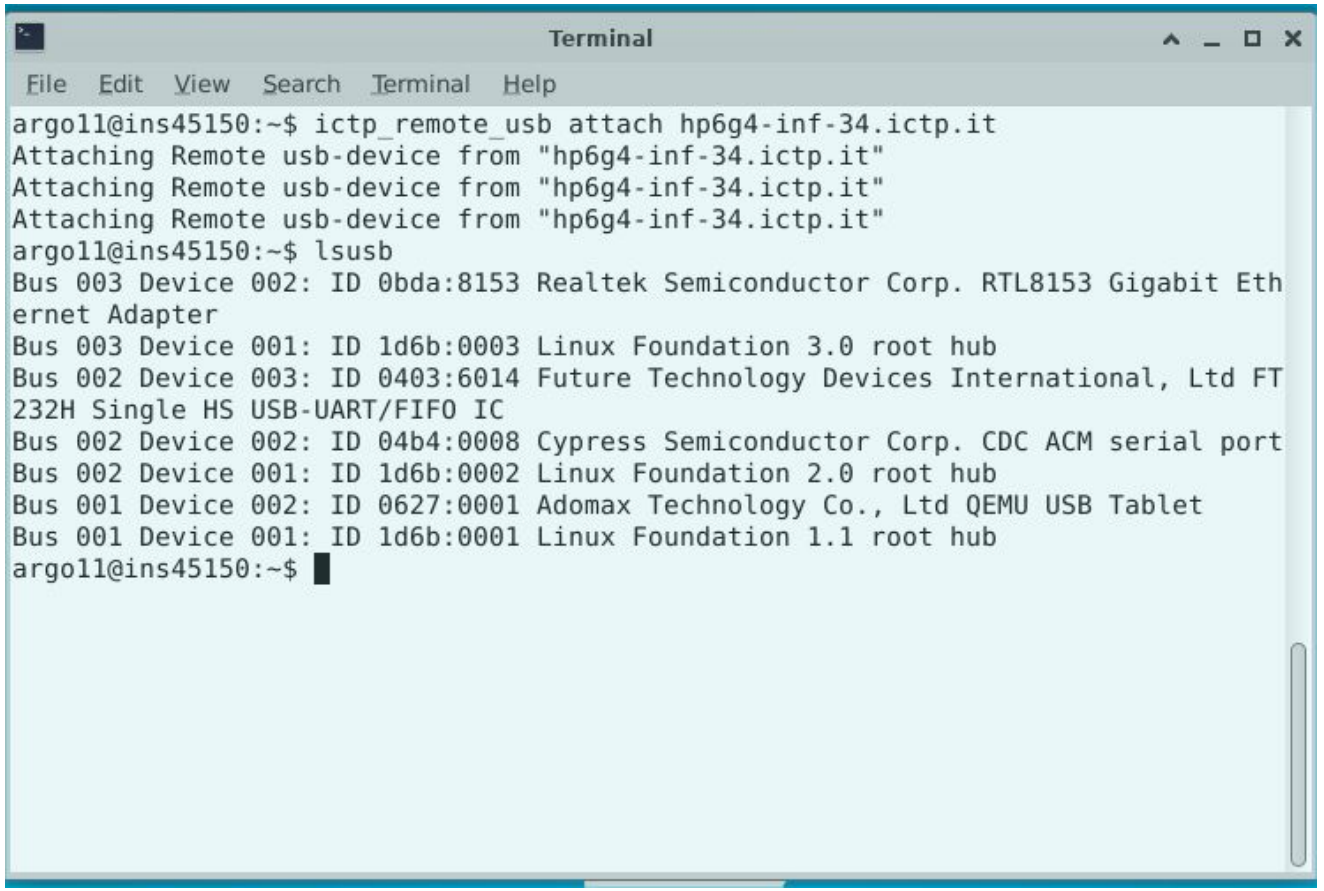
On the Virtual Machine

```
ictp_remote_usb attach <local_machine_name>.ictp.it
```

Connecting to The Zedboard (VM)

On the Virtual Machine e.g.

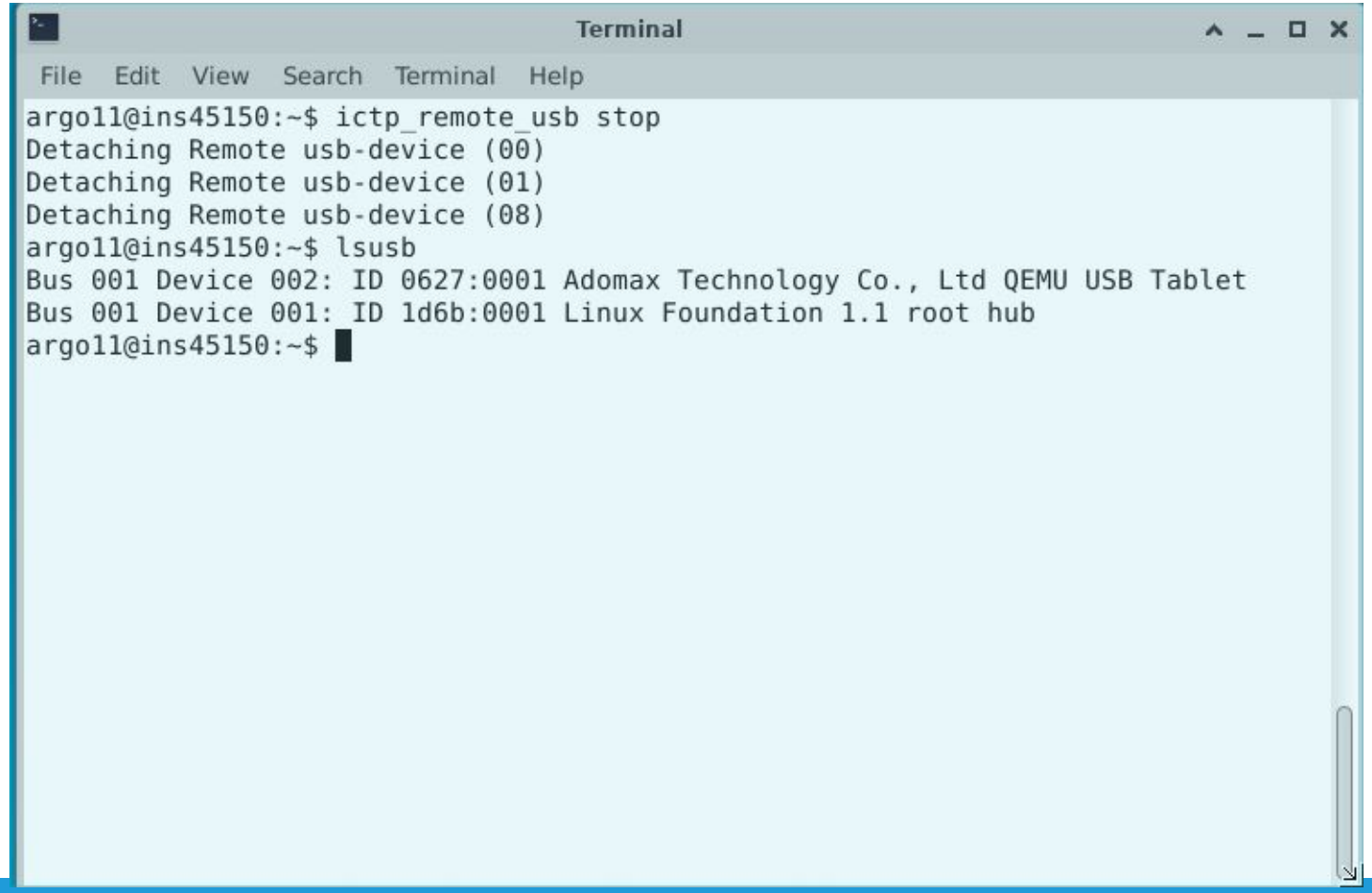
```
ictp_remote_usb attach hp6g4-inf-XX.ictp.it
```



Connecting to The Zedboard (VM)

On the Virtual Machine e.g.

```
ictp_remote_usb stop
```



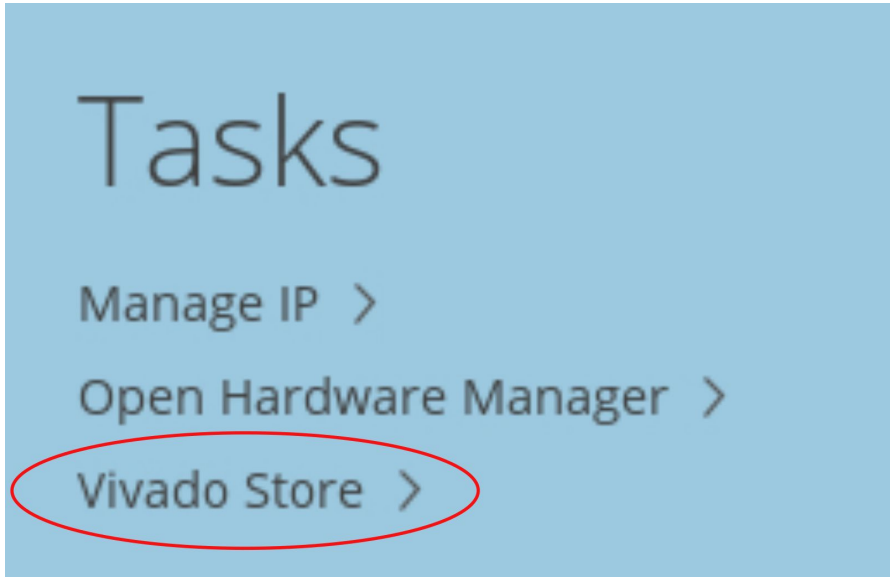
Connecting to The Zedboard (VM)

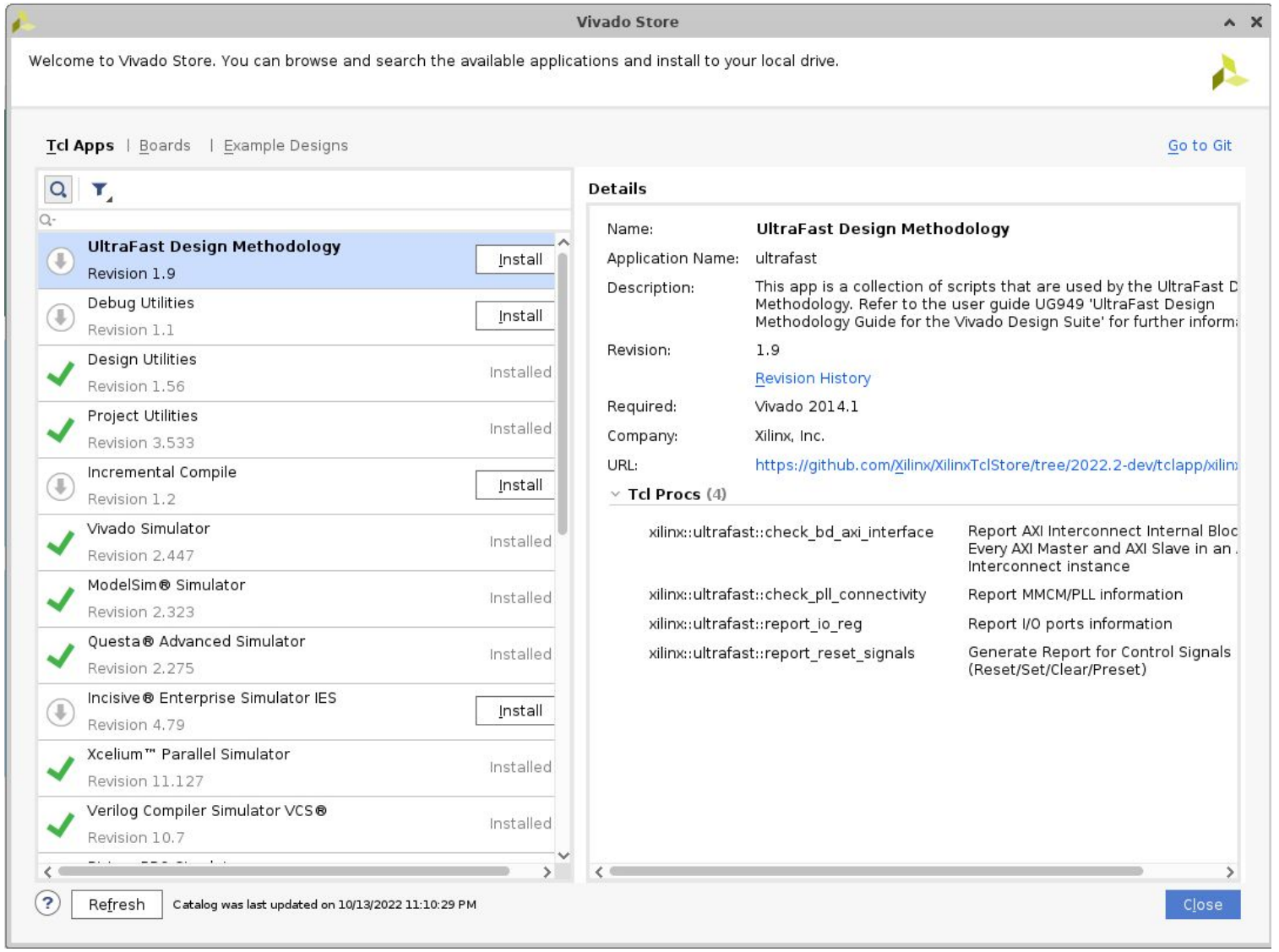
Let your neighbor connect to the Zedboard

```
ictp_remote_usb attach <local_machine_name>.ictp.it
```

```
ictp_remote_usb stop
```

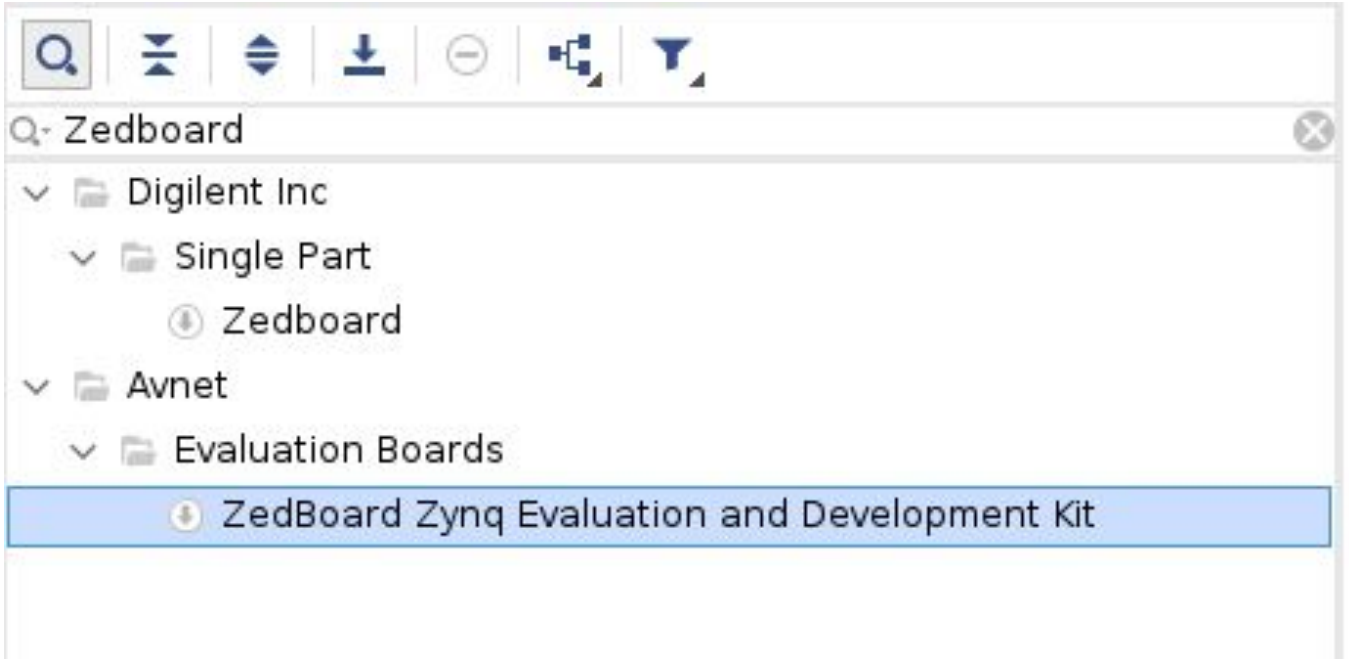
Testing the Hardware: Installing the Zedboard





Testing the Hardware: Installing the Zedboard

Tcl Apps | **Boards** | Example Designs



Name: **ZedBoard Zynq Evaluation and Development Kit**

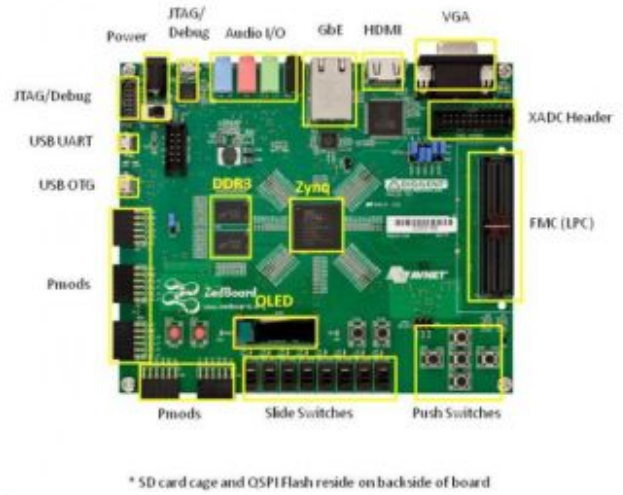
Description: ZedBoard Zynq Evaluation and Development Kit

Revision: 1.4

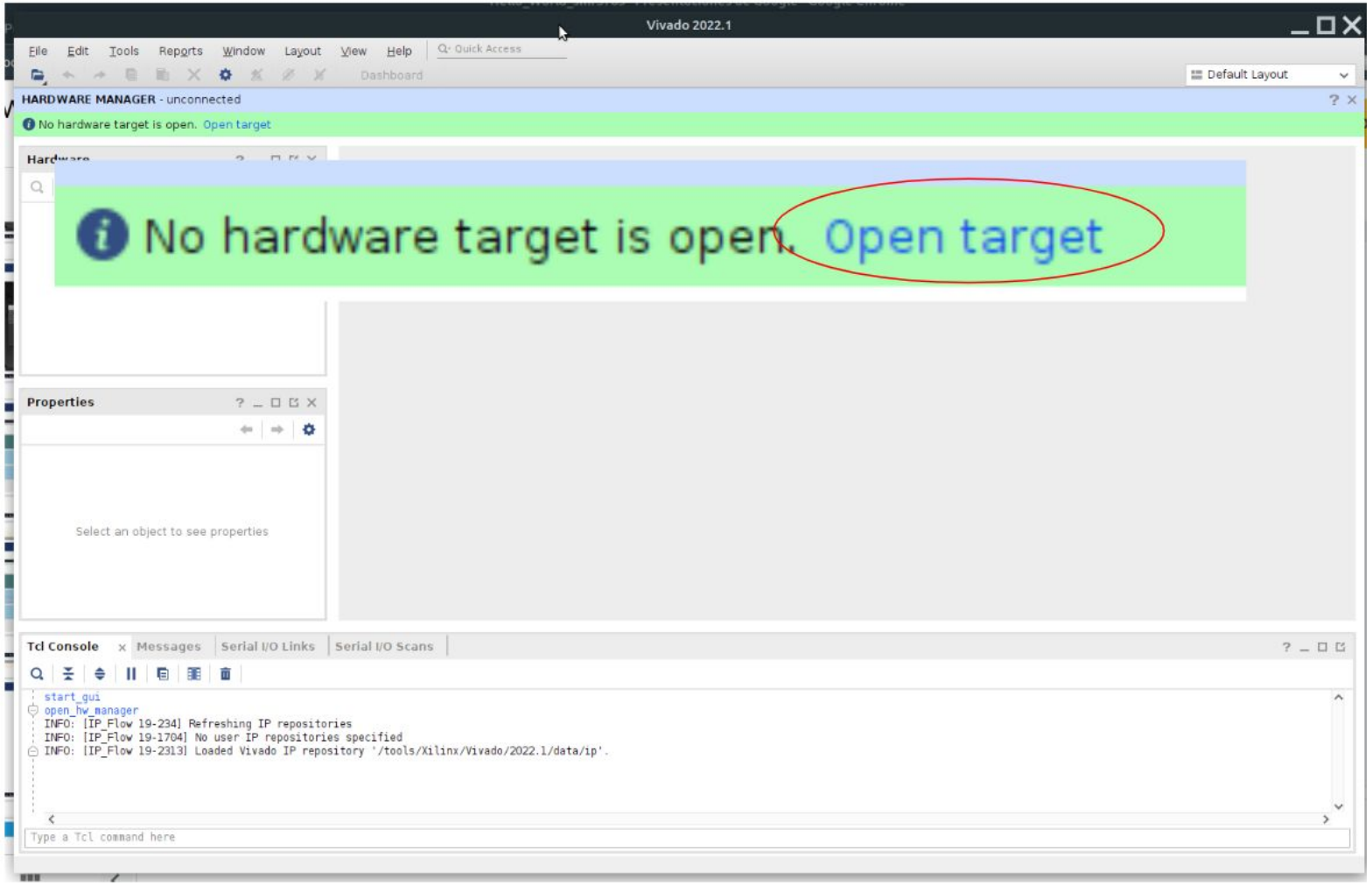
Revision History

URL: https://github.com/Xilinx/XilinxBoardStore/tree/2022.2/boards/Avnet/zedb

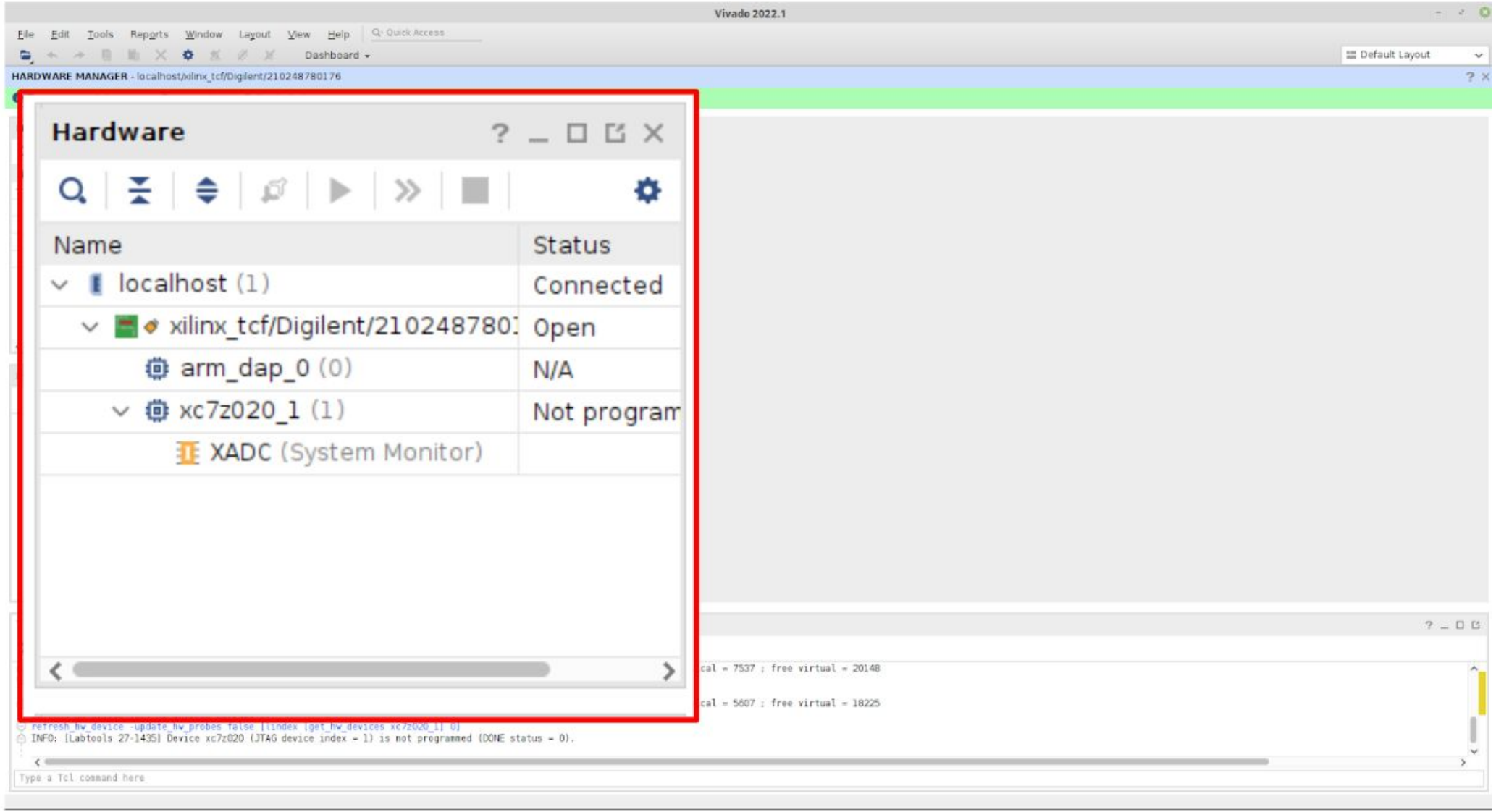
Company: Avnet



Testing the Hardware: Checking the connection



Testing the Hardware: Checking the connection



Testing the Hardware: Checking the UART