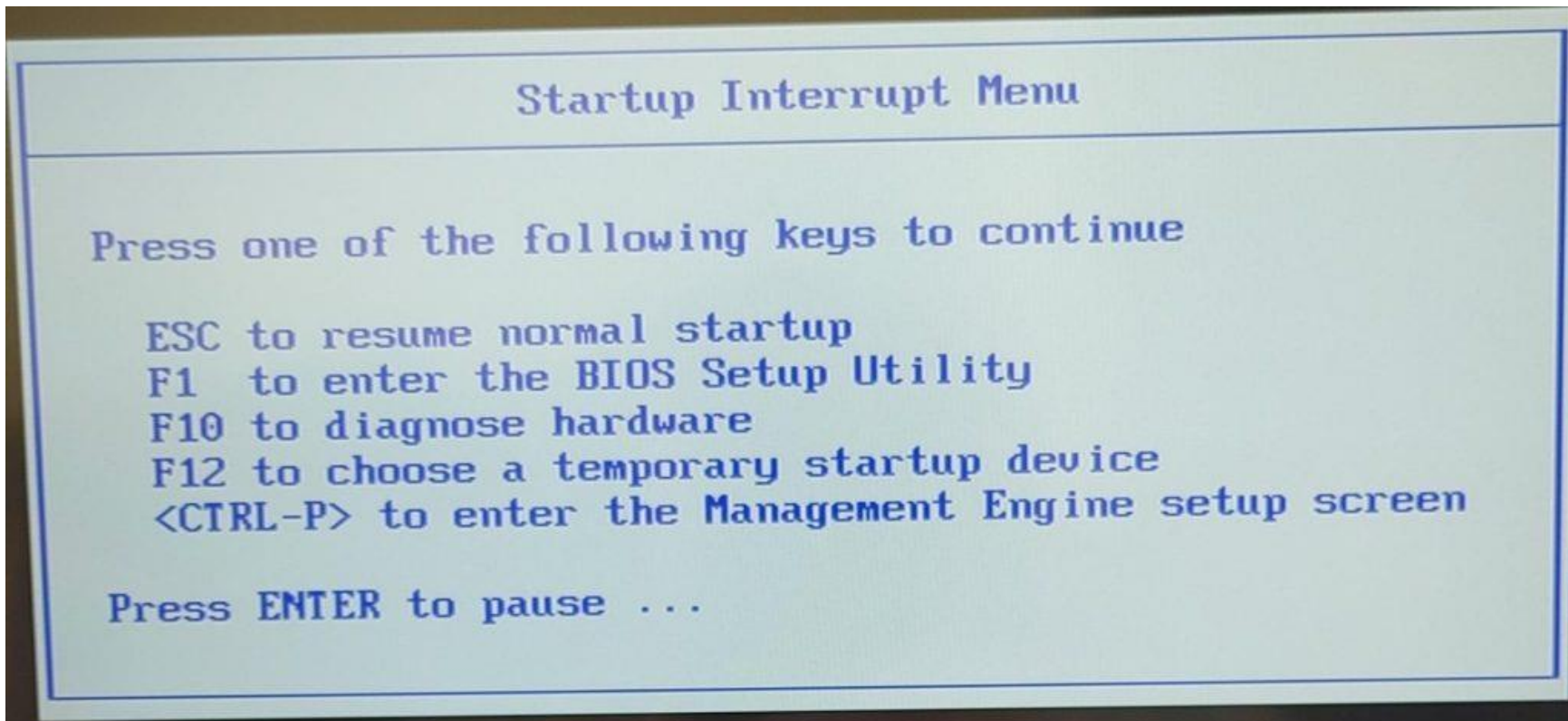
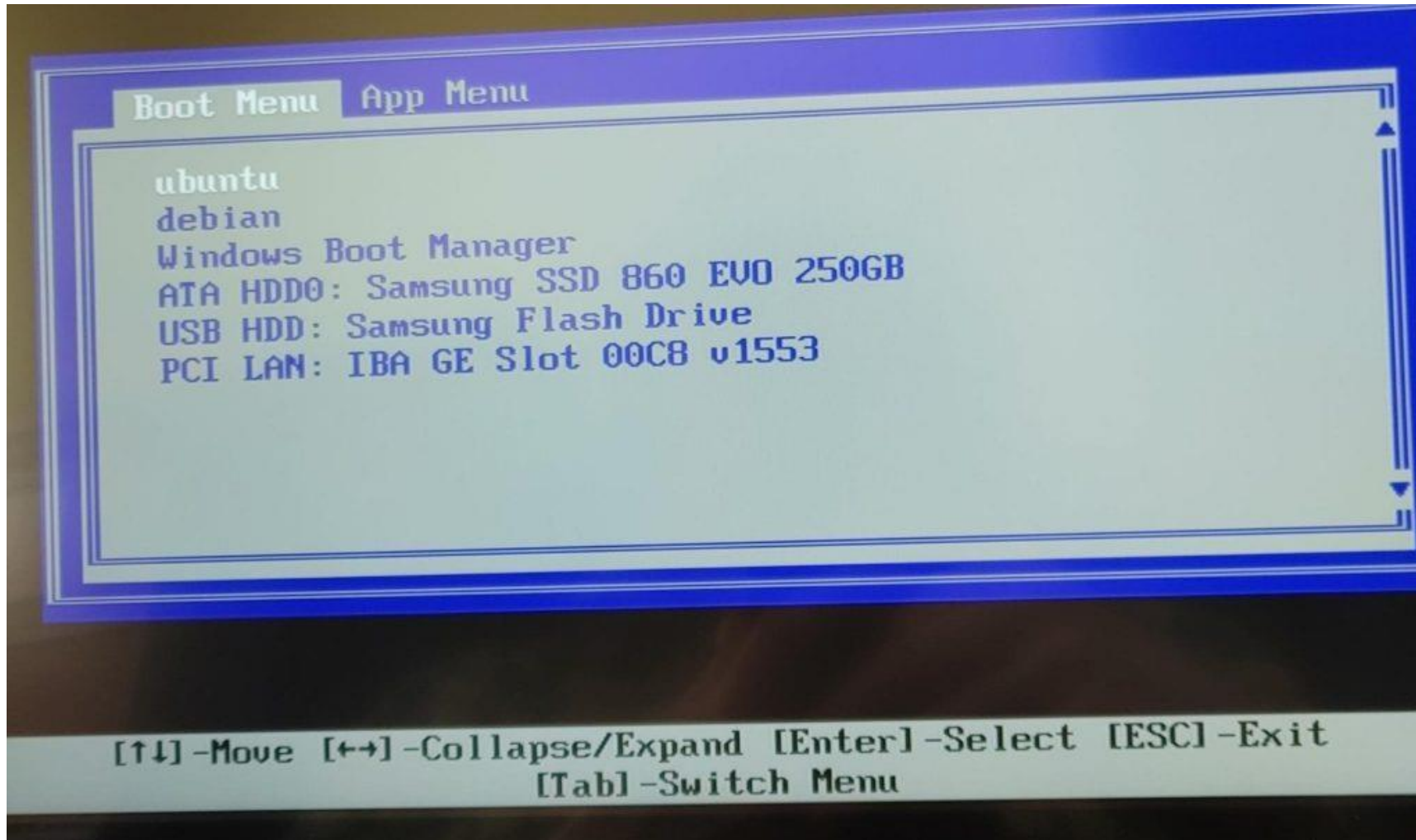


Edger thumb drive users start here. Regular Linux users start at [this later step below](#).

With the computer powered off install the Edger thumb drive in the computer. Turn the power on and enter the keyboard key to break into the BIOS. This is typically Enter for Thinkpad laptops but might be F2 or DEL or another key depending on the computer type. A screen similar to the one below will be shown.



Enter F12 or whatever is needed for your system to get to the boot menu like this:



Select the “Samsung Flash Drive” (typically with arrow and enter key) to boot the thumb drive. The system will come up with a regular Ubuntu desktop screen (cat with whiskers). Now proceed with instructions below with [this link](#).

Start here to install Edger on a regular Linux system.

Use a web browser to go to <https://github.com/TriEmbed/edger>, then navigate to the raw linux_install.sh file (shortcut [here](#)). Right click the “Raw” button on the right:

v.15.02.2023

The screenshot displays a web browser window with the GitHub repository for `TriEmbed/edger` open. The URL in the address bar is `https://github.com/TriEmbed/edger/blob/main/linux_install.sh`. The repository page shows the file `linux_install.sh` selected, with a commit history and a list of contributors. The file content is displayed in a code editor, showing a bash script that sets up the environment for the `edger` project. The script includes comments and instructions for installing dependencies and running the project. To the right of the code editor, a terminal window is open, showing the execution of the script. The terminal output includes the command `less /tmp/install.sh`, which displays the script content, and the command `mkdir $HOME/Pictures/edger-instructions`, which creates the directory.

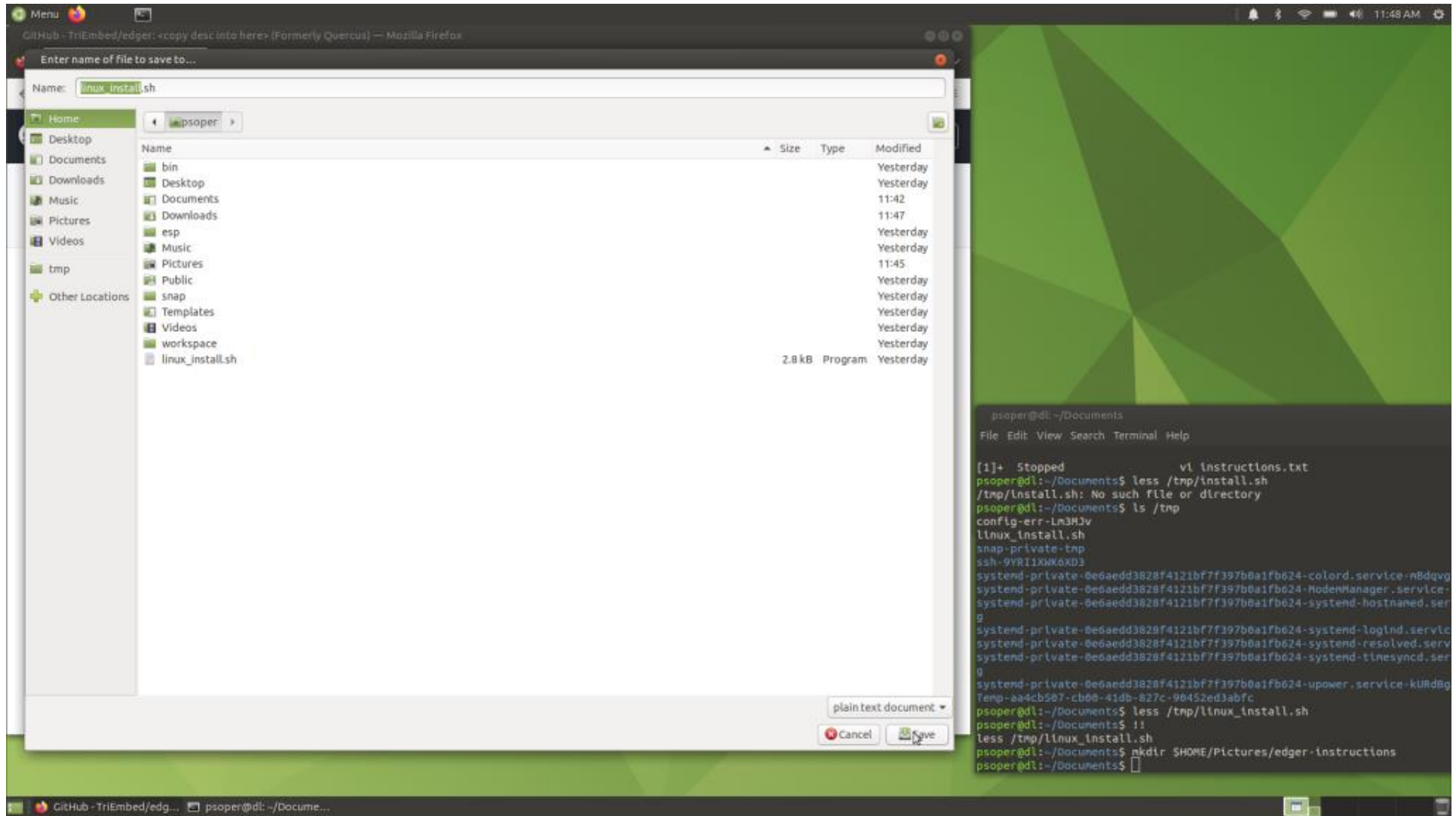
```
1 #!/bin/bash
2 # Hack Edger onto a Linux system. Currently just handles Ubuntu.
3
4 # Here is an important todo: This script needs to detect changes to the edger repo and rebuild ant and pardvark as needed
5
6 # The Espressif does have the recipes for package installs for other linux flavors
7
8 echo "====="
9 echo $PATH $HOME/.bashrc | grep "USER/bin"
10 echo "====="
11
12 if [ $? -ne 0 ] ; then
13     echo '$HOME/bin is not in your search rules in .bashrc: adding PATH=\"$PATH:$HOME/bin'
14     echo 'export PATH=\"$PATH:$HOME/bin' >>$HOME/.bashrc
15     echo "run this script again"
16     exit 0
17 fi
18
19 groups | grep " dialout"
20 if [ $? -ne 0 ] ; then
```

```
psoper@dl: ~/Documents
File Edit View Search Terminal Help

[1]+  Stopped                  vl instructions.txt
psoper@dl:~/Documents$ less /tmp/install.sh
/tmp/install.sh: No such file or directory
psoper@dl:~/Documents$ ls /tmp
config-err-Ln3H3v
linux_install.sh
snap-private-tmp
ssh-9YR11XWkX0D1
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-color.service-n8dqvg
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-ModemManager.service-
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-systemd-hostnamed.ser
0
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-systemd-logind.servic
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-systemd-resolved.serv
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-systemd-timesyncd.ser
0
systemd-private-0e6aedd3828f4121bf7f397b0a1fb624-upower.service-kURd8g
Temp-aa4cb507-cb00-41db-827c-90452ed3abfc
psoper@dl:~/Documents$ less /tmp/linux_install.sh
psoper@dl:~/Documents$ !!
psoper@dl:~/Documents$ mkdir $HOME/Pictures/edger-instructions
psoper@dl:~/Documents$
```

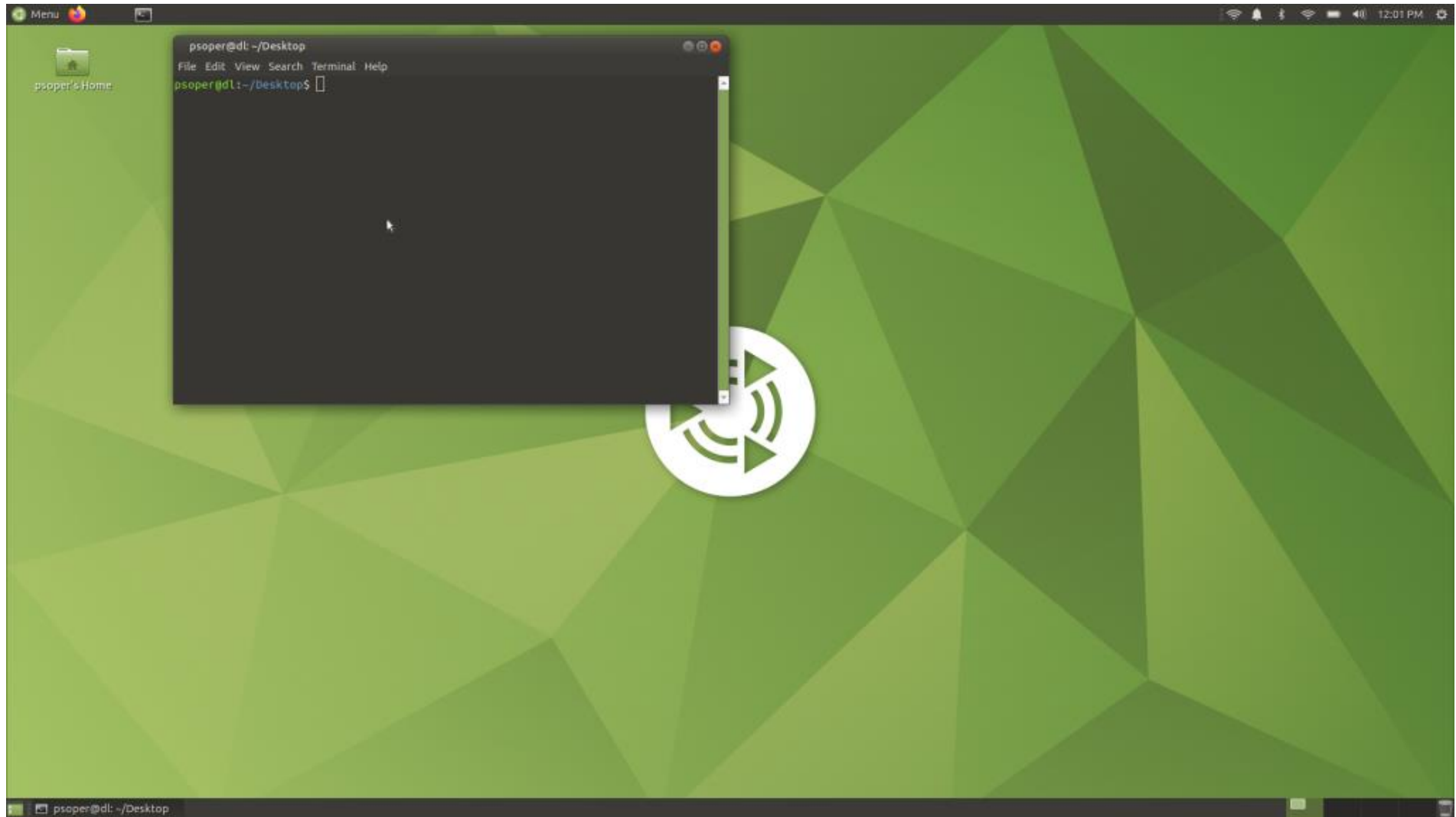
Hover over “Raw” and right click and select “save link as” and save this file to `linux_install.sh` in your home directory:

v.15.02.2023



Start shell session with right click in the desktop window and then left click on “open in terminal”:

v.15.02.2023



Get to home directory with a cd command with no argument:

cd

Run the install script:

bash linux_install.sh

v.15.02.2023

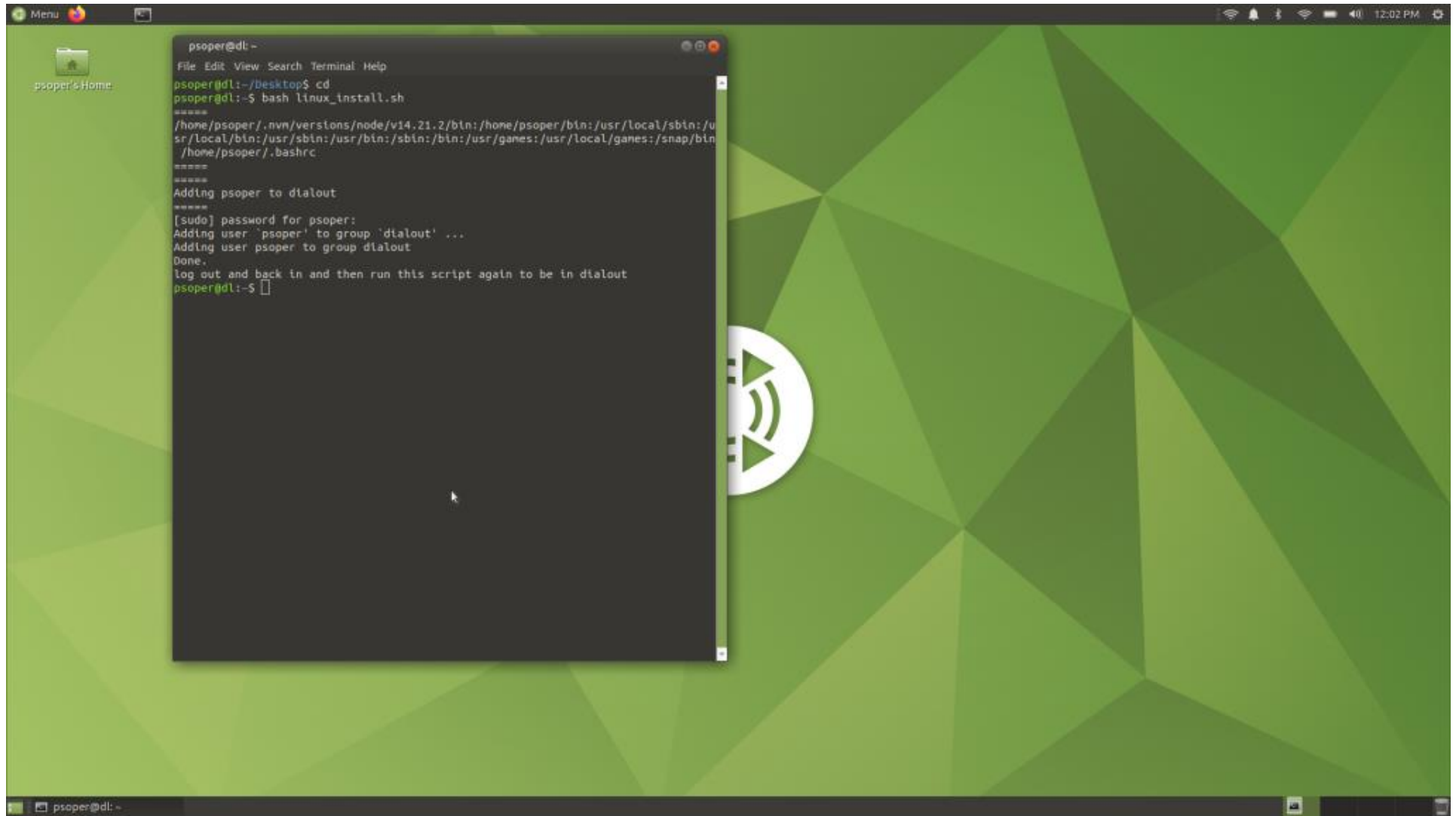
Enter your password to allow the sudo command within the script. The first time the script is executed the following directories are created (\$HOME stands for your home directory pathname):

\$HOME/workspace \$HOME/workspace/esp32 \$HOME/esp \$HOME/bin

In the case of \$HOME/bin this directory is present already with Ubuntu Mate but has to be created for vanilla Ubuntu and .bashrc has to be edited automatically by the script. For the latter situation the script will stop with instructions to start a new terminal session before restarting the script so that the new search rules are available to the script. Do this by simply exiting the terminal session (“X” in upper right corner of terminal window) and then using the “open in terminal” operation again and again typing “cd” and then “bash linux_install.sh”.

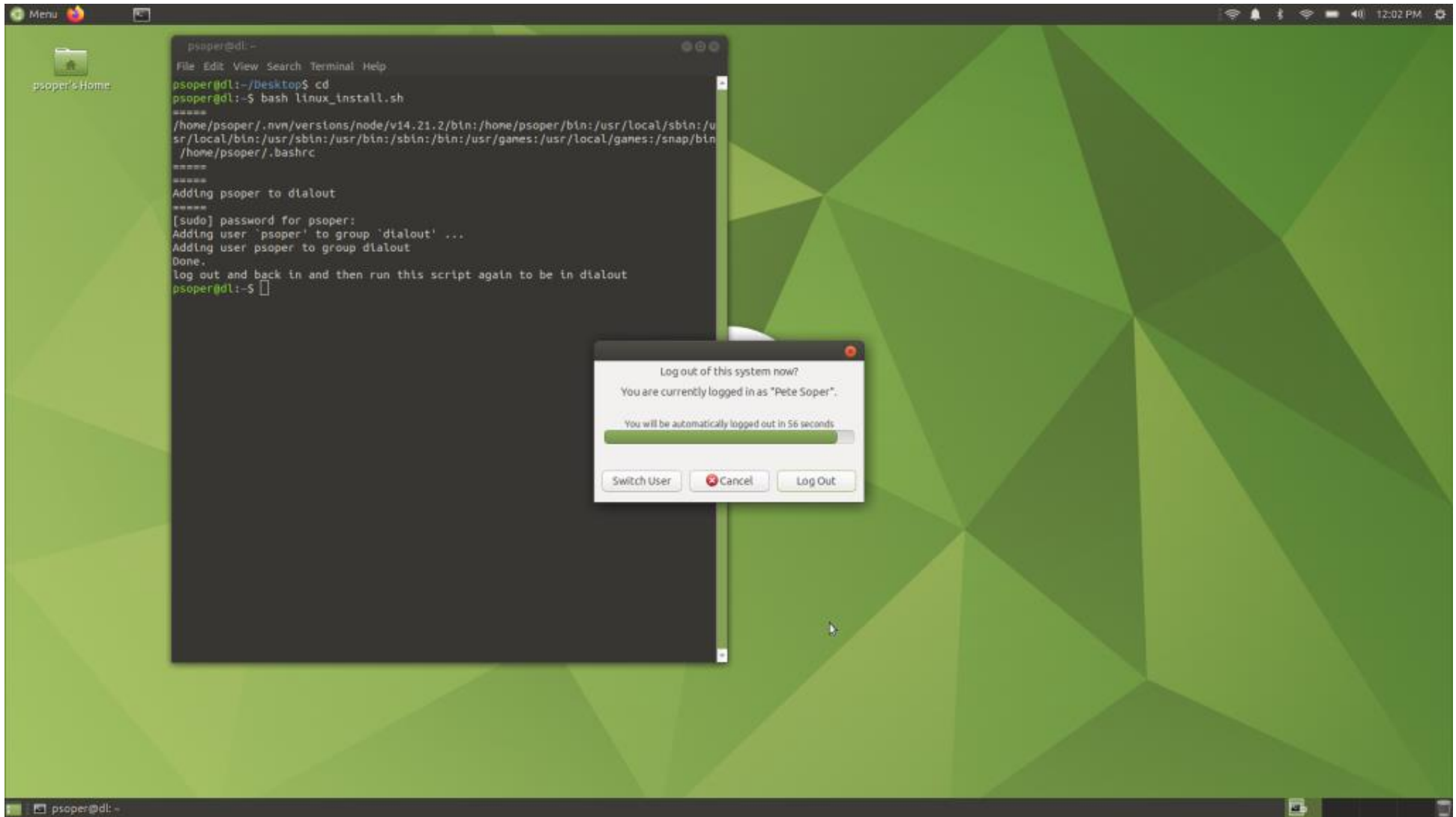
The Edger repository is cloned as \$HOME/workspace/esp32/edger and your user id is added to the “dialout” Linux group to allow access to the development board via USB. You must follow instructions to log out and back in:

v.15.02.2023



Use the icon in the extreme upper right of the screen (a cog wheel or tiny monitor typically) with left clicks to log out:

v.15.02.2023

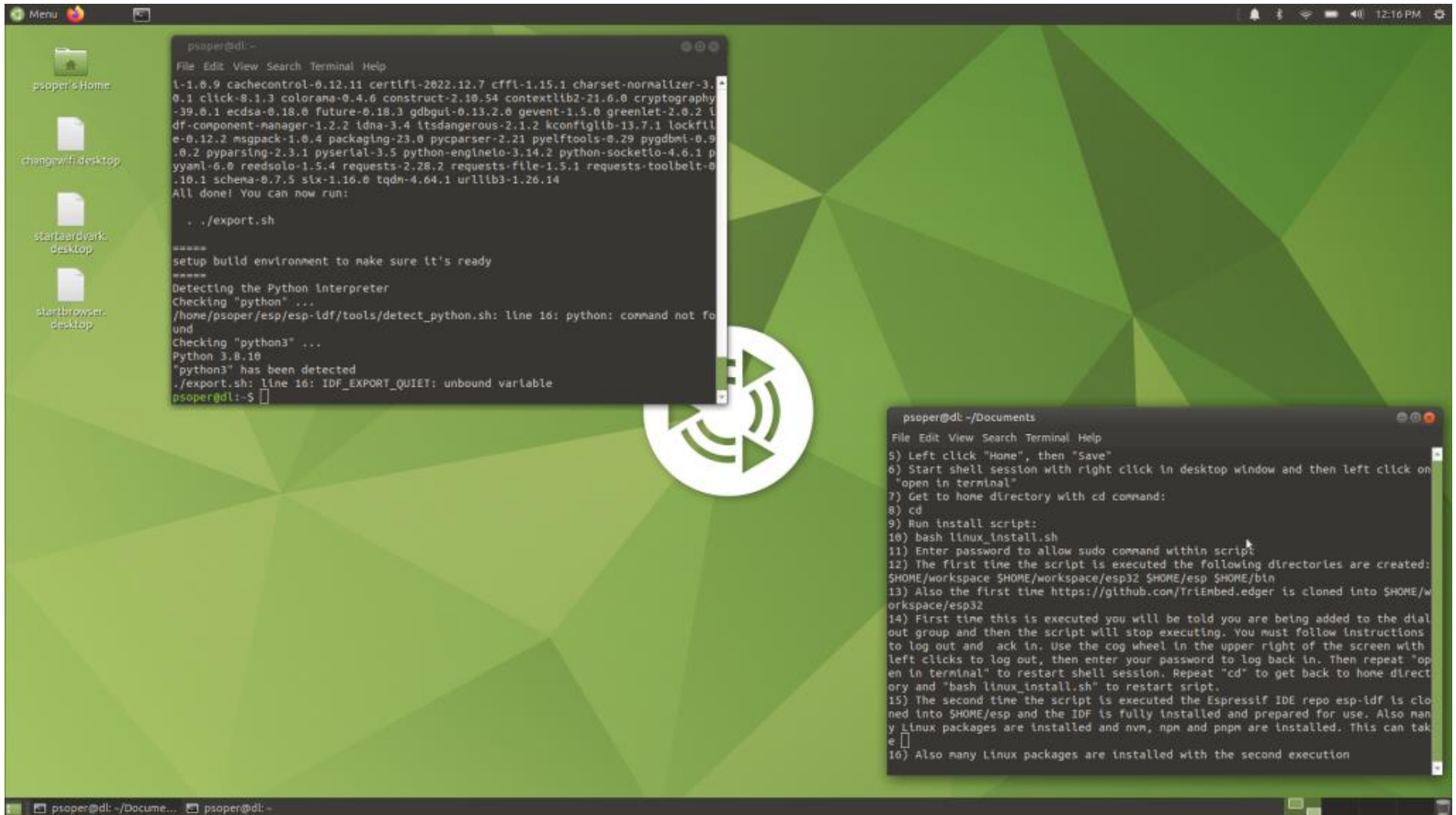


Then enter your password to log back in. Then repeat “open in terminal” to restart a shell session, repeat “cd” to get back to your home directory and “bash linux_install.sh” to restart the script. Once this logout/login is done once it will never need to be done again.

The next time the script is executed the Espressif IDE repo esp-idf is cloned into \$HOME/esp and the IDF is partially or fully installed and prepared for use. Also many Linux packages are installed and nvm, npm and pnpm are installed. This can all take many minutes to complete depending on Internet speed and the speed of the computer.

v.15.02.2023

This script execution may not fully install the Espressif IDF depending on some Linux specifics. If the output ends this way the “bash linux_install.sh” command must be repeated one more time by :

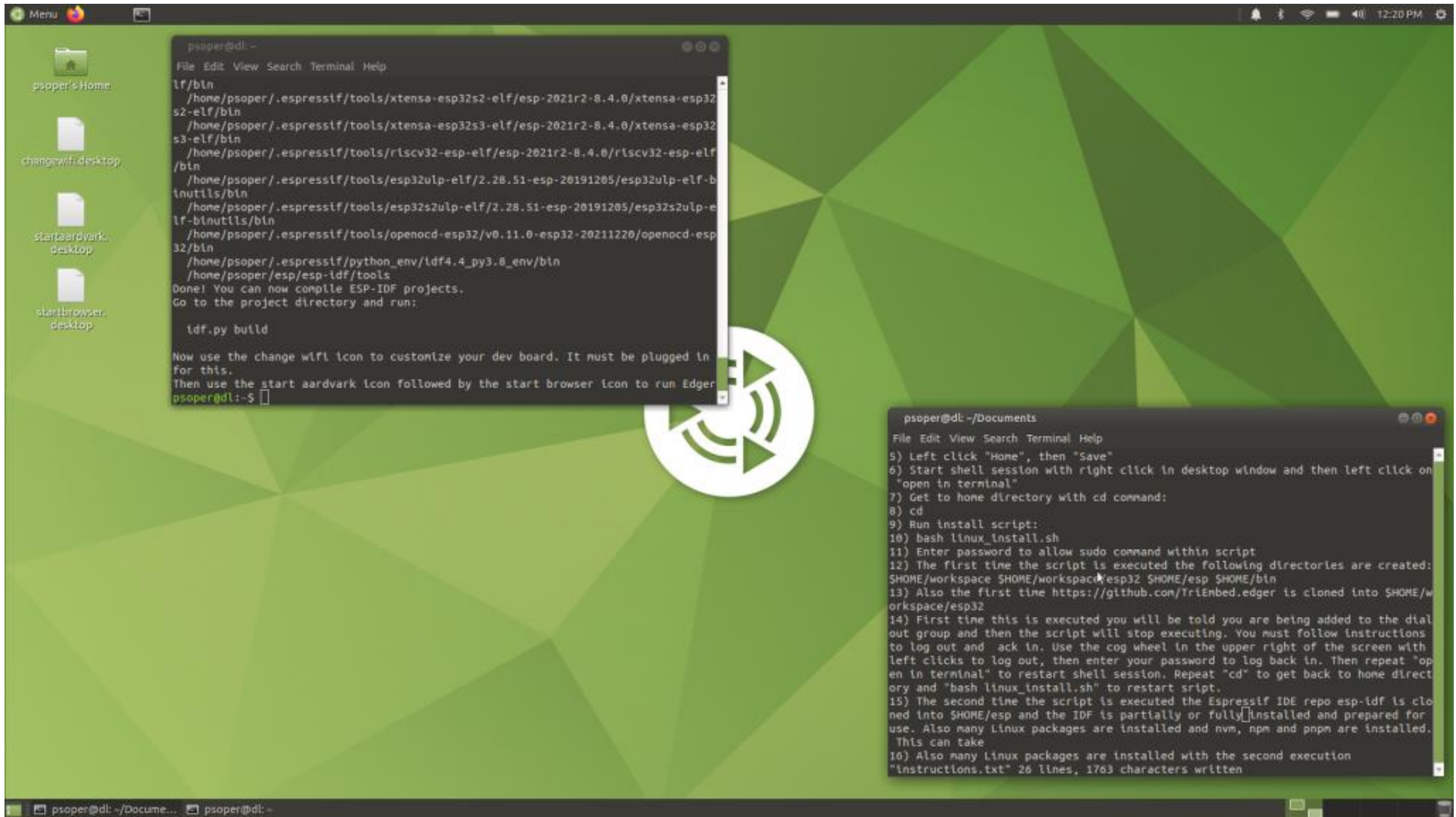


```
psoper@dl: ~  
File Edit View Search Terminal Help  
i-1.0.9 cachecontrol-0.12.11 certifi-2022.12.7 cffi-1.15.1 charset-normalizer-3.  
0.1 click-8.1.3 colorama-0.4.6 construct-2.10.54 contextlib2-21.0.0 cryptography  
-39.0.1 ecdsa-0.18.0 future-0.18.3 gdbgui-0.13.2.0 gevent-1.5.0 greenlet-2.0.2 i  
df-component-manager-1.2.2 idna-3.4 itsdangerous-2.1.2 kconfiglib-13.7.1 lockfil  
e-0.12.2 msgpack-1.0.4 packaging-23.0 pycparser-2.21 pyelftools-0.29 pygdbmi-0.9  
.0.2 pyparsing-2.3.1 pyserial-3.5 python-engineio-3.14.2 python-socketio-4.6.1 p  
yyaml-6.0 reedsolo-1.5.4 requests-2.28.2 requests-file-1.5.1 requests-toolbelt-0  
.10.1 schena-0.7.5 six-1.16.0 tqdm-4.64.1 urllib3-1.26.14  
All done! You can now run:  
  
./export.sh  
  
=====  
setup build environment to make sure it's ready  
=====  
Detecting the Python Interpreter  
Checking "python" ...  
/home/psoper/esp/esp-idf/tools/detect_python.sh: line 16: python: command not fo  
und  
Checking "python3" ...  
Python 3.8.10  
"python3" has been detected  
./export.sh: line 16: IDF_EXPORT_QUIET: unbound variable  
psoper@dl:~$
```

```
psoper@dl: ~/Documents  
File Edit View Search Terminal Help  
5) Left click "Home", then "Save"  
6) Start shell session with right click in desktop window and then left click on  
"open in terminal"  
7) Get to home directory with cd command:  
8) cd  
9) Run install script:  
10) bash linux_install.sh  
11) Enter password to allow sudo command within script  
12) The first time the script is executed the following directories are created:  
$HOME/workspace $HOME/workspace/esp32 $HOME/esp $HOME/bin  
13) Also the first time https://github.com/TriEnbed.edger is cloned into $HOME/w  
orkspace/esp32  
14) First time this is executed you will be told you are being added to the dial  
out group and then the script will stop executing. You must follow instructions  
to log out and back in. Use the cog wheel in the upper right of the screen with  
left clicks to log out, then enter your password to log back in. Then repeat "op  
en in terminal" to restart shell session. Repeat "cd" to get back to home direct  
ory and "bash linux_install.sh" to restart script.  
15) The second time the script is executed the Espressif IDE repo esp-idf is clon  
ed into $HOME/esp and the IDF is fully installed and prepared for use. Also man  
y Linux packages are installed and nvm, npm and pnpm are installed. This can tak  
e  
16) Also many Linux packages are installed with the second execution
```

When the script execution completes normally the output will look like this:

v.15.02.2023



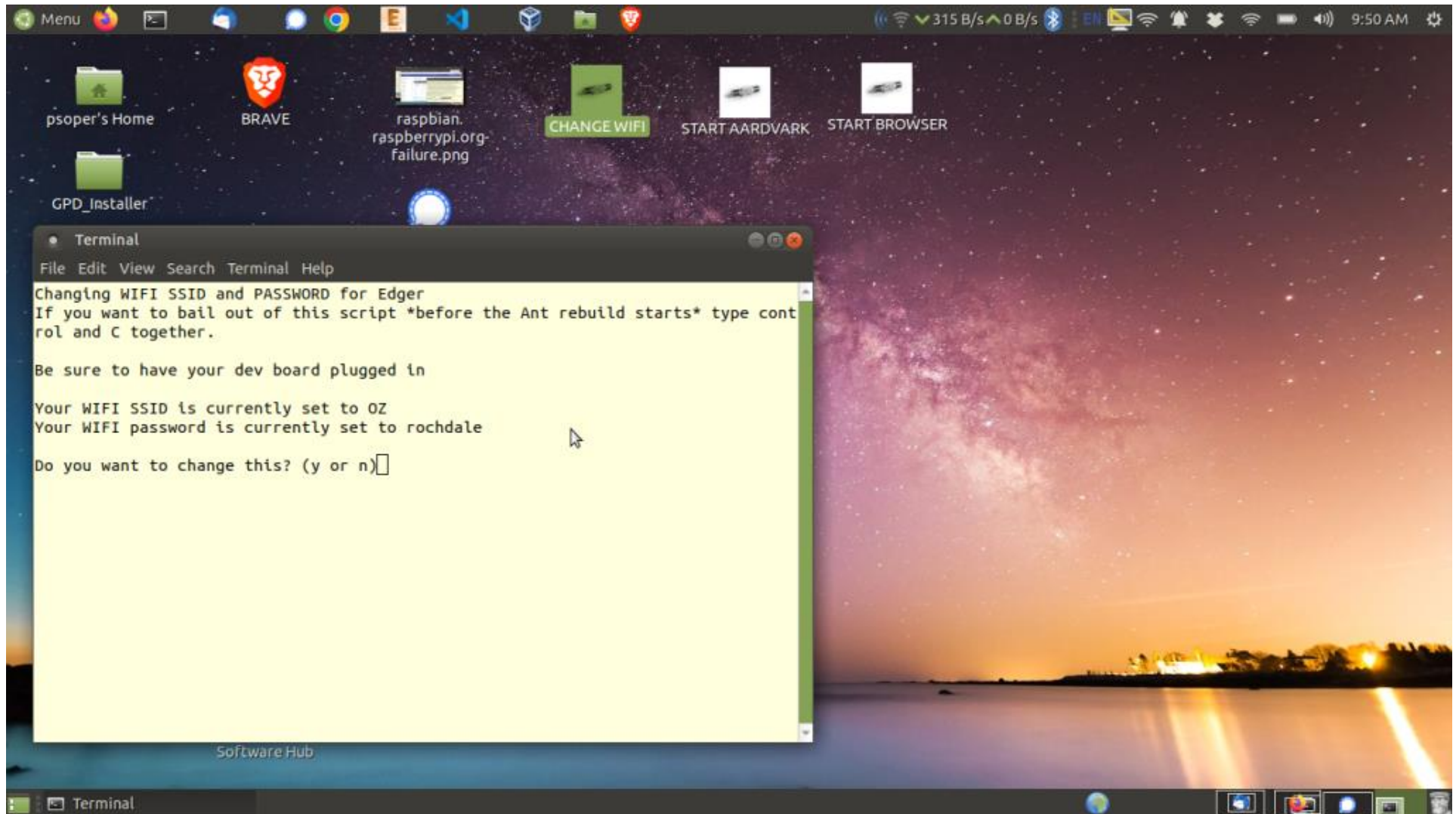
Thumb drive users resume here.

Notice that three new desktop icons are present: “CHANGE WIFI”, “START AARDVARK”, and “START BROWSER” (these should all have “Edger” in them with the next installer update!). These three launcher icons are all that are needed to use Edger and they are used in order, but “CHANGE WIFI” is ordinarily only needed one time.

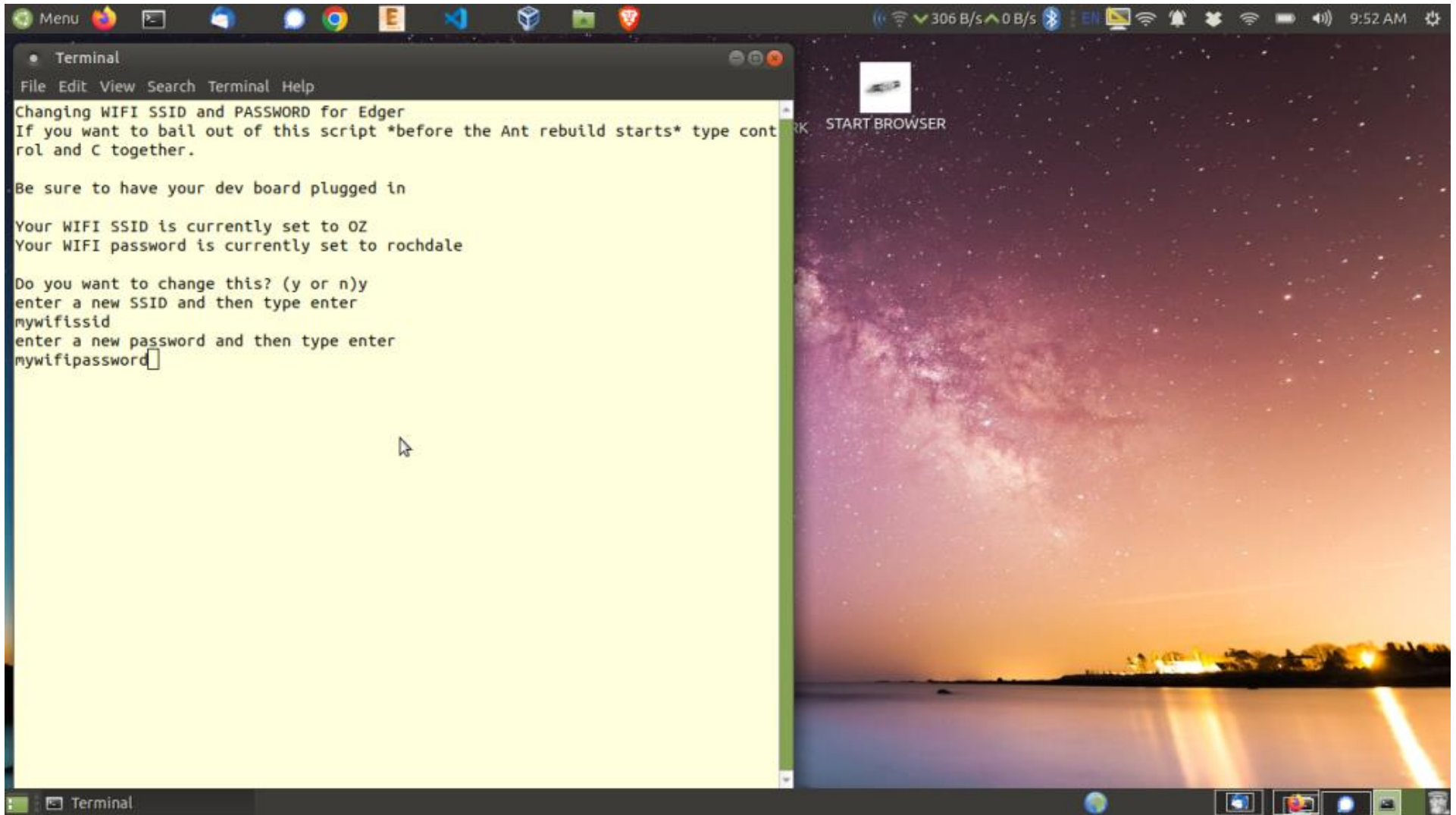
v.15.02.2023

To prepare to run Edger plug in your version two white development board using the USB connector on the end of the board (not the one pointing sideways).

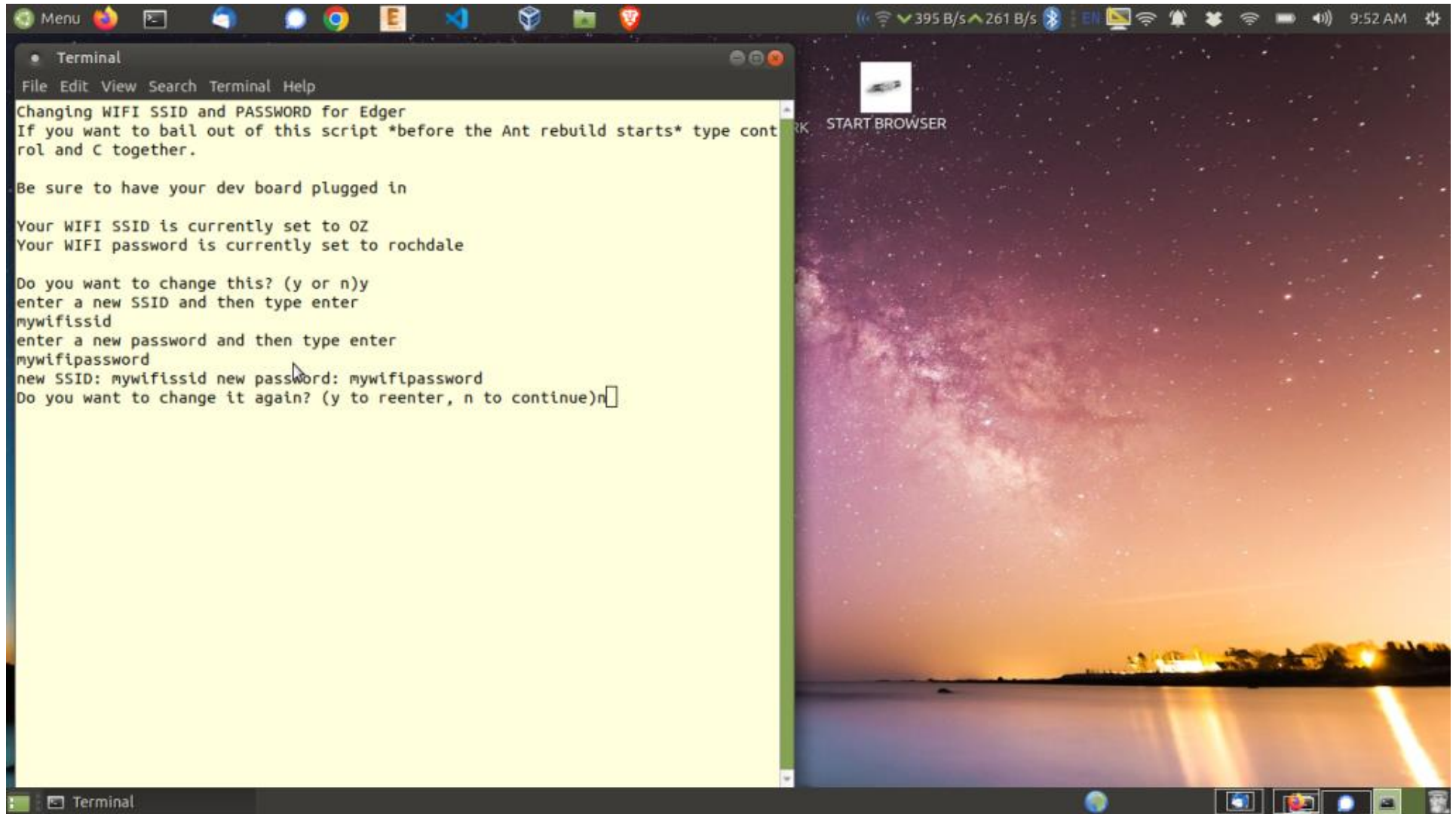
If you haven't already done so, double left click the "CHANGE WIFI" icon to configure your local WIFI connection. Again, the Edger dev board **MUST** be plugged in for this and subsequent steps. Follow the instructions for entering your WIFI SSID and password. The Edger "Ant" firmware will be rebuilt and flashed into your dev board:



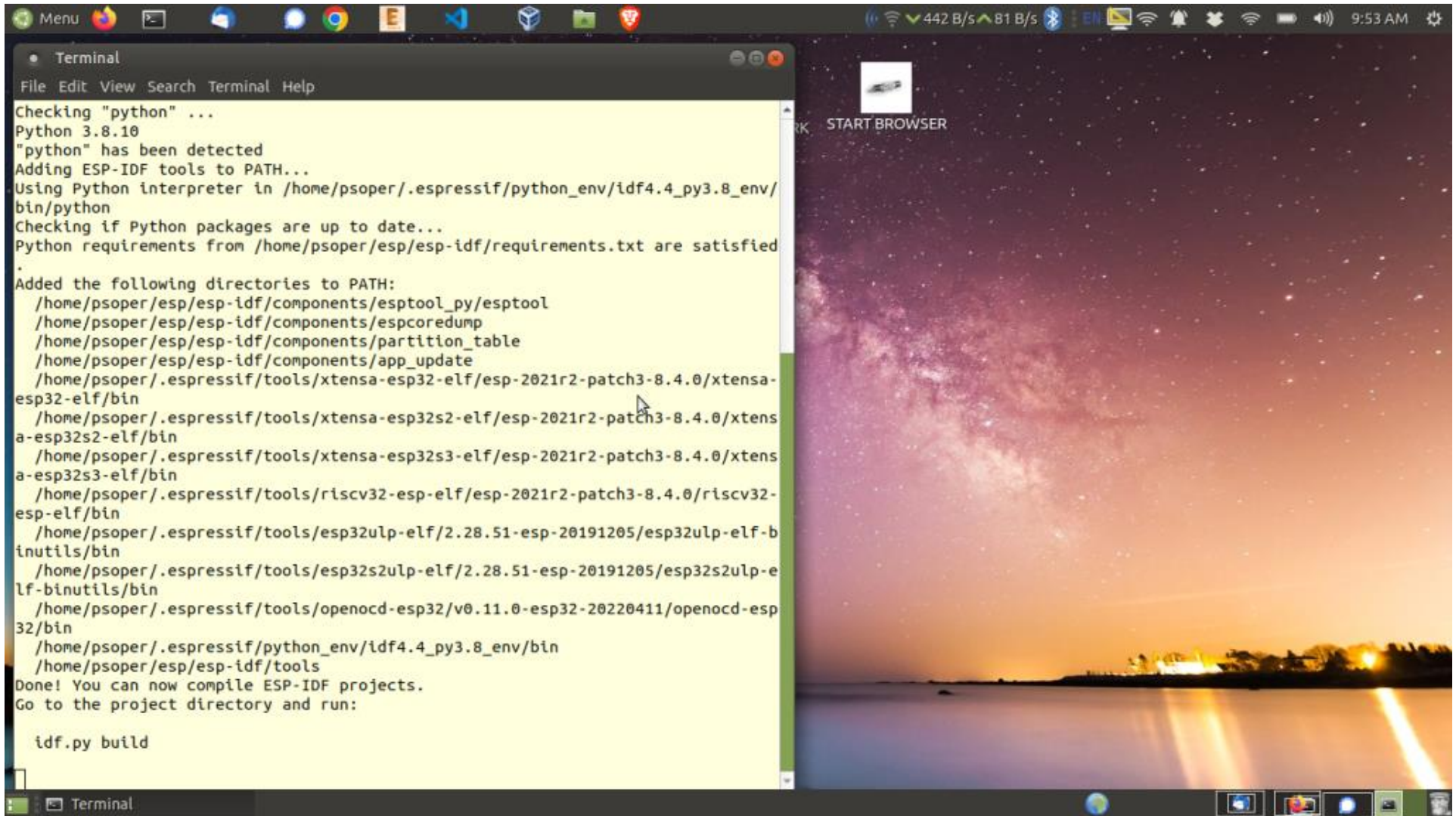
v.15.02.2023



v.15.02.2023



v.15.02.2023



The screenshot shows a Linux desktop environment. The top panel includes a 'Menu' button, application icons for Firefox, a terminal, and several system status icons (network, battery, Bluetooth, etc.) on the right. The time is 9:53 AM. A terminal window is open, displaying the output of the 'idf.py build' command. The output shows that Python 3.8.10 is detected and the ESP-IDF tools are added to the PATH. A list of directories is added to the PATH, including various toolchain and component paths. The terminal also shows a 'START BROWSER' button on the desktop.

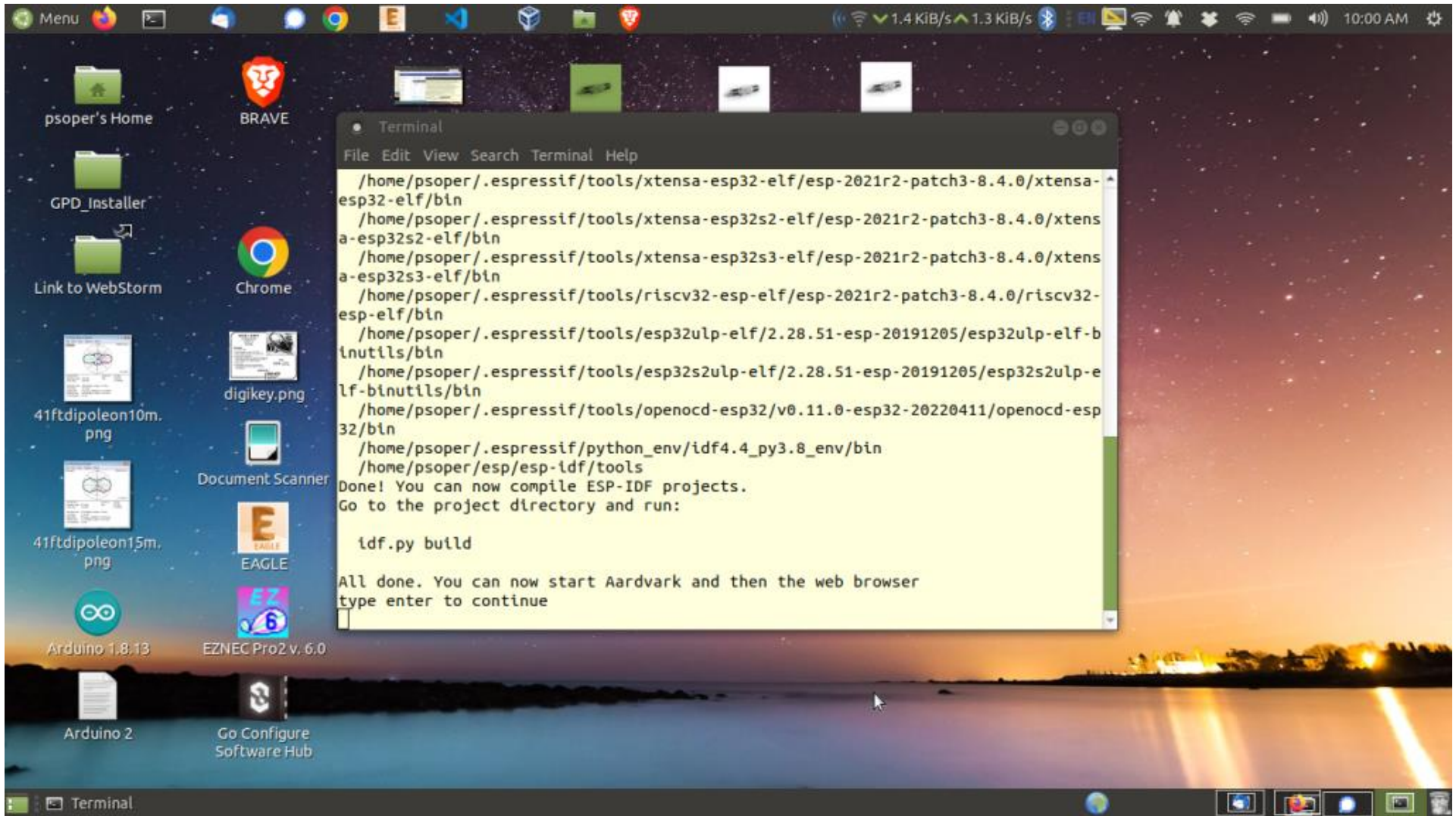
```
Terminal
File Edit View Search Terminal Help
Checking "python" ...
Python 3.8.10
"python" has been detected
Adding ESP-IDF tools to PATH...
Using Python interpreter in /home/psoper/.espressif/python_env/idf4.4_py3.8_env/
bin/python
Checking if Python packages are up to date...
Python requirements from /home/psoper/esp/esp-idf/requirements.txt are satisfied
.
Added the following directories to PATH:
/home/psoper/esp/esp-idf/components/esptool_py/esptool
/home/psoper/esp/esp-idf/components/espcoredump
/home/psoper/esp/esp-idf/components/partition_table
/home/psoper/esp/esp-idf/components/app_update
/home/psoper/.espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-
esp32-elf/bin
/home/psoper/.espressif/tools/xtensa-esp32s2-elf/esp-2021r2-patch3-8.4.0/xtens
a-esp32s2-elf/bin
/home/psoper/.espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch3-8.4.0/xtens
a-esp32s3-elf/bin
/home/psoper/.espressif/tools/riscv32-esp-elf/esp-2021r2-patch3-8.4.0/riscv32-
esp-elf/bin
/home/psoper/.espressif/tools/esp32ulp-elf/2.28.51-esp-20191205/esp32ulp-elf-b
inutils/bin
/home/psoper/.espressif/tools/esp32s2ulp-elf/2.28.51-esp-20191205/esp32s2ulp-e
lf-binutils/bin
/home/psoper/.espressif/tools/openocd-esp32/v0.11.0-esp32-20220411/openocd-esp
32/bin
/home/psoper/.espressif/python_env/idf4.4_py3.8_env/bin
/home/psoper/esp/esp-idf/tools
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build
```

WAIT FOR THIS SCREEN TO UPDATE. It will take a few minutes. (There should be an information line something like “Ant is rebuilding: please wait” but unfortunately there is none). Next you will see:

The most likely error message at this point will be the result of not having a dev board plugged in. (Note a rev 1 dev board might “just work”. Please try this if you can and let us know your results). Normal completion will look like this:

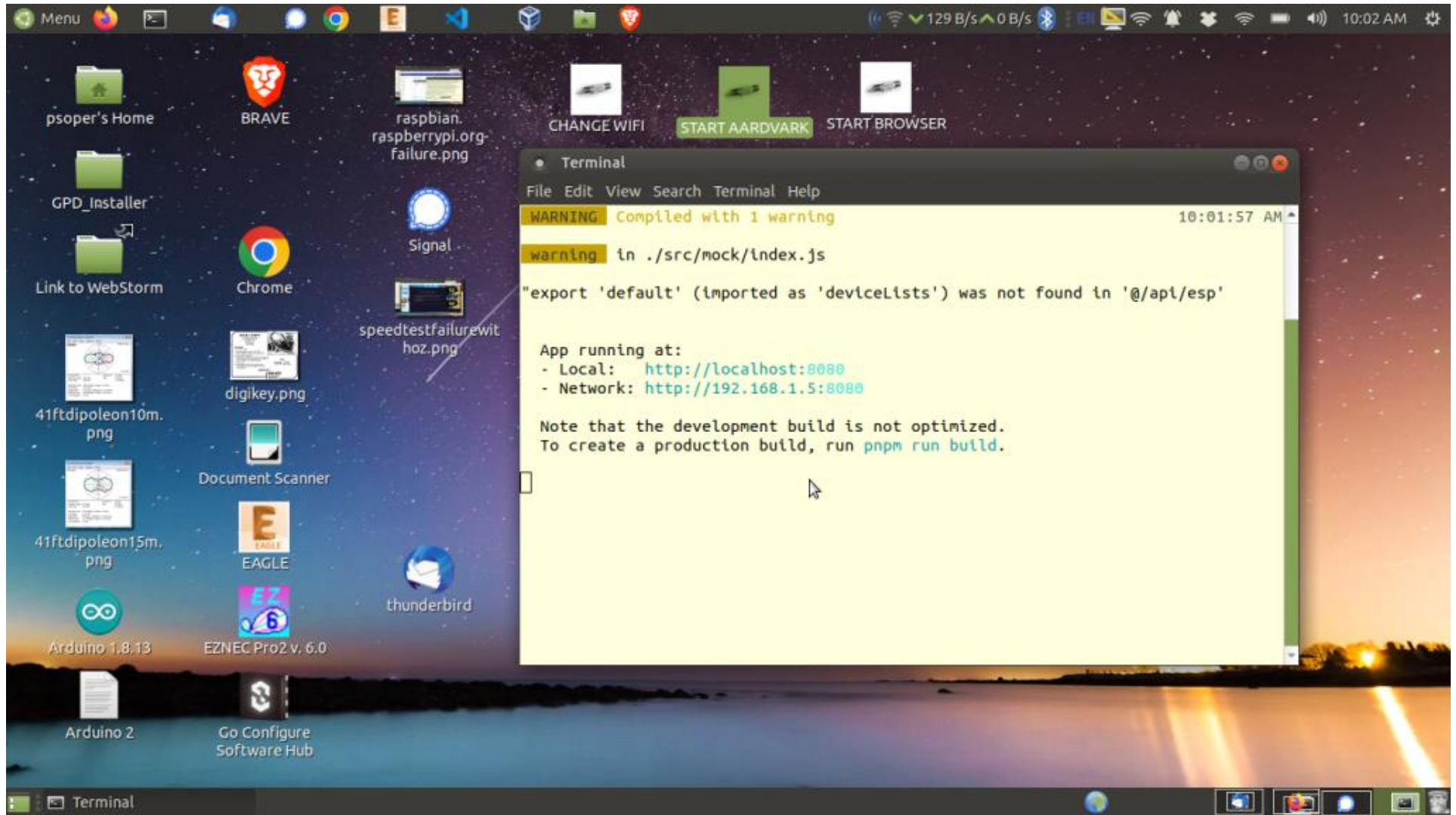
v.15.02.2023



Next type enter to dismiss this window.

Now double left click the “START AARDVARK” icon. This will be your ordinary starting point going forward. You will see some fast changing text and then the Aardvark software will be running in a “Node JS” session, ready for connection with your browser:

v.15.02.2023



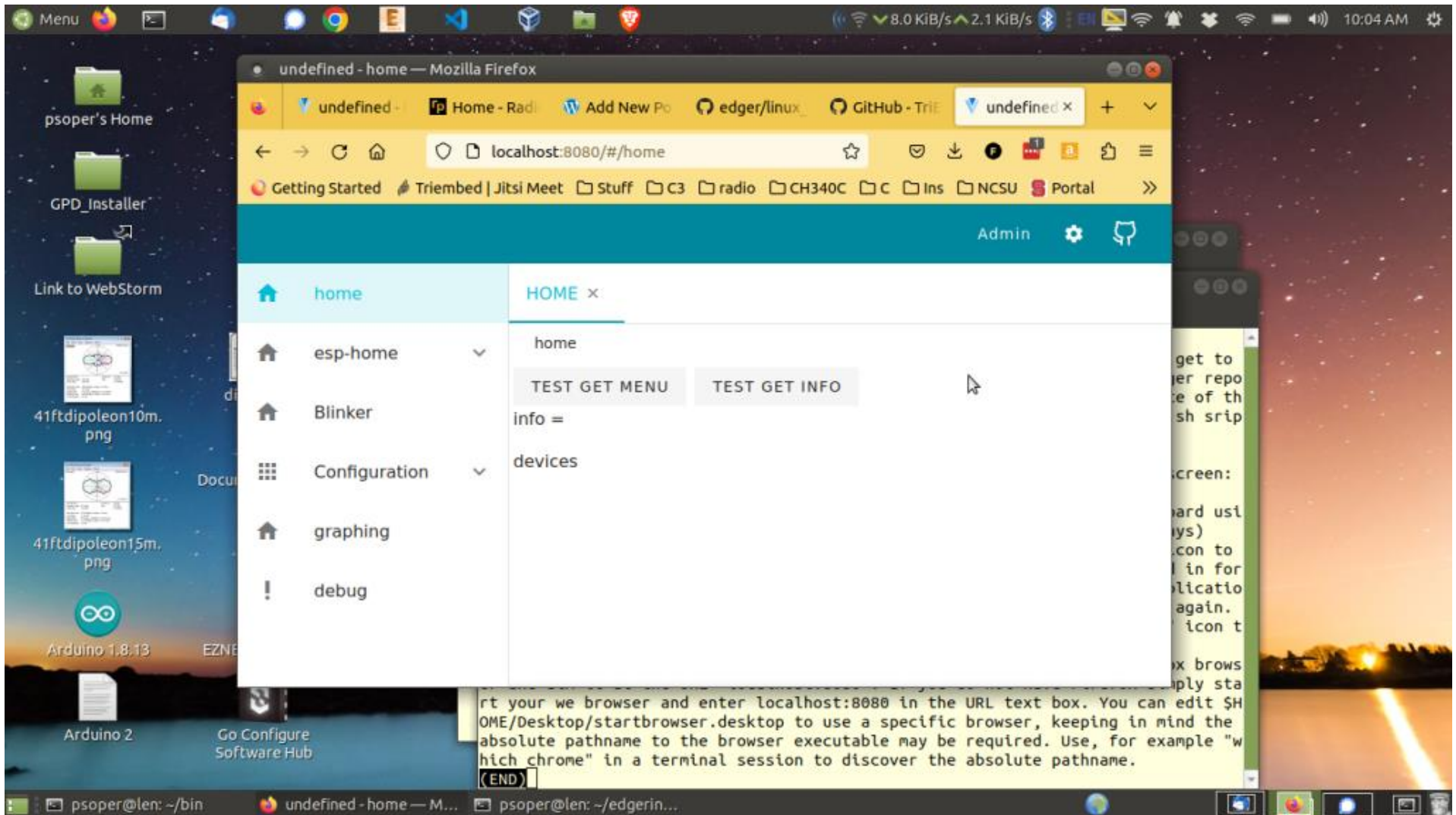
Leave this window up while using Edger. When you don't need it anymore you can dismiss it by typing "control" and "C" in the window (after left clicking to get focus).

Now double left click the "START BROWSER" icon. This will open the Firefox browser and aim it at the URL "localhost:8080". If you do not have Firefox simply start your web browser and enter localhost:8080 in the URL text box. You can edit \$HOME/Desktop/startbrowser.desktop to use a

v.15.02.2023

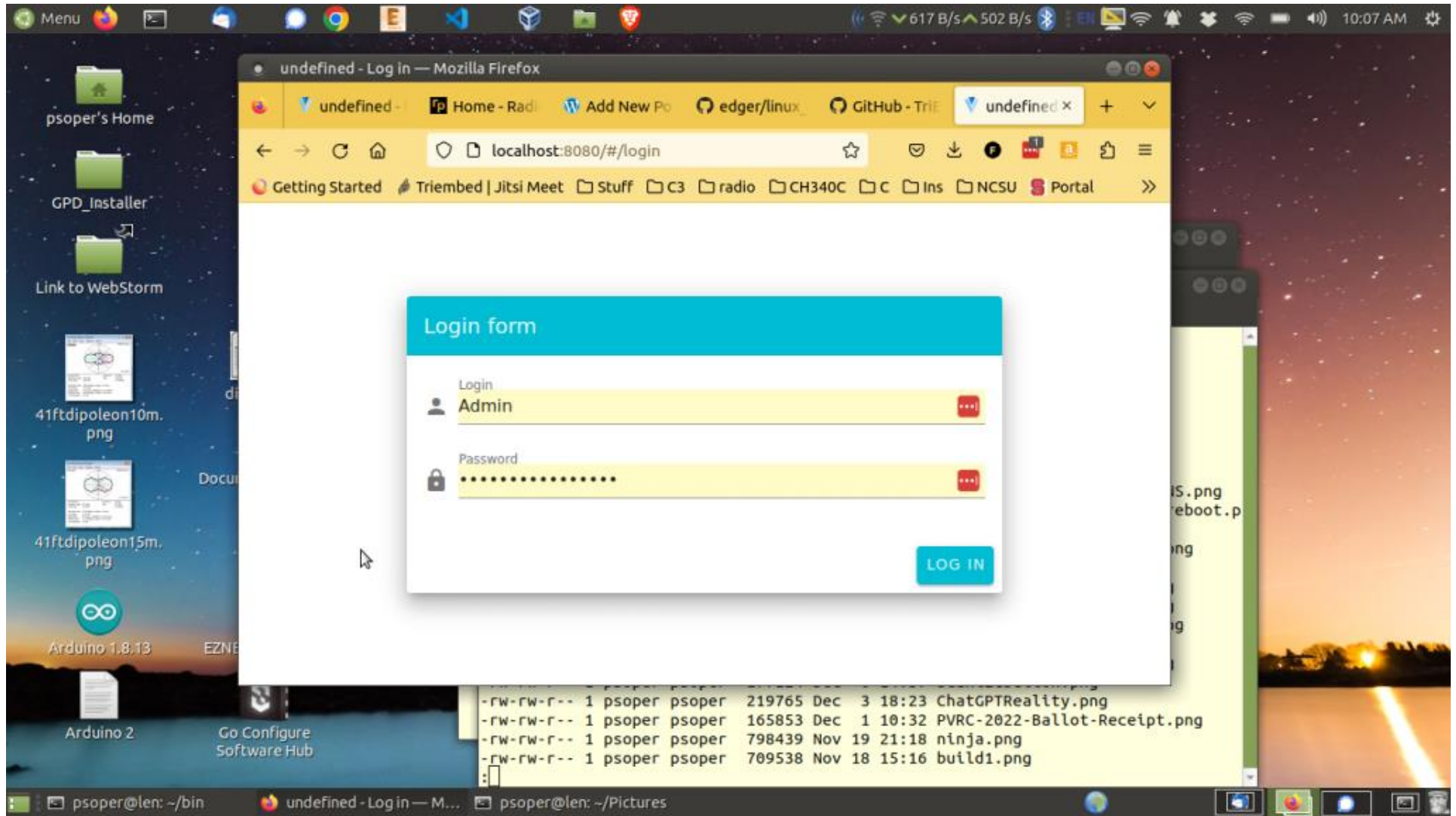
specific browser, keeping in mind the absolute pathname to the browser executable may be required. Use, for example “which chrome” in a terminal session to discover the absolute pathname.

The Edger Aardvark software will now be running in your browser:



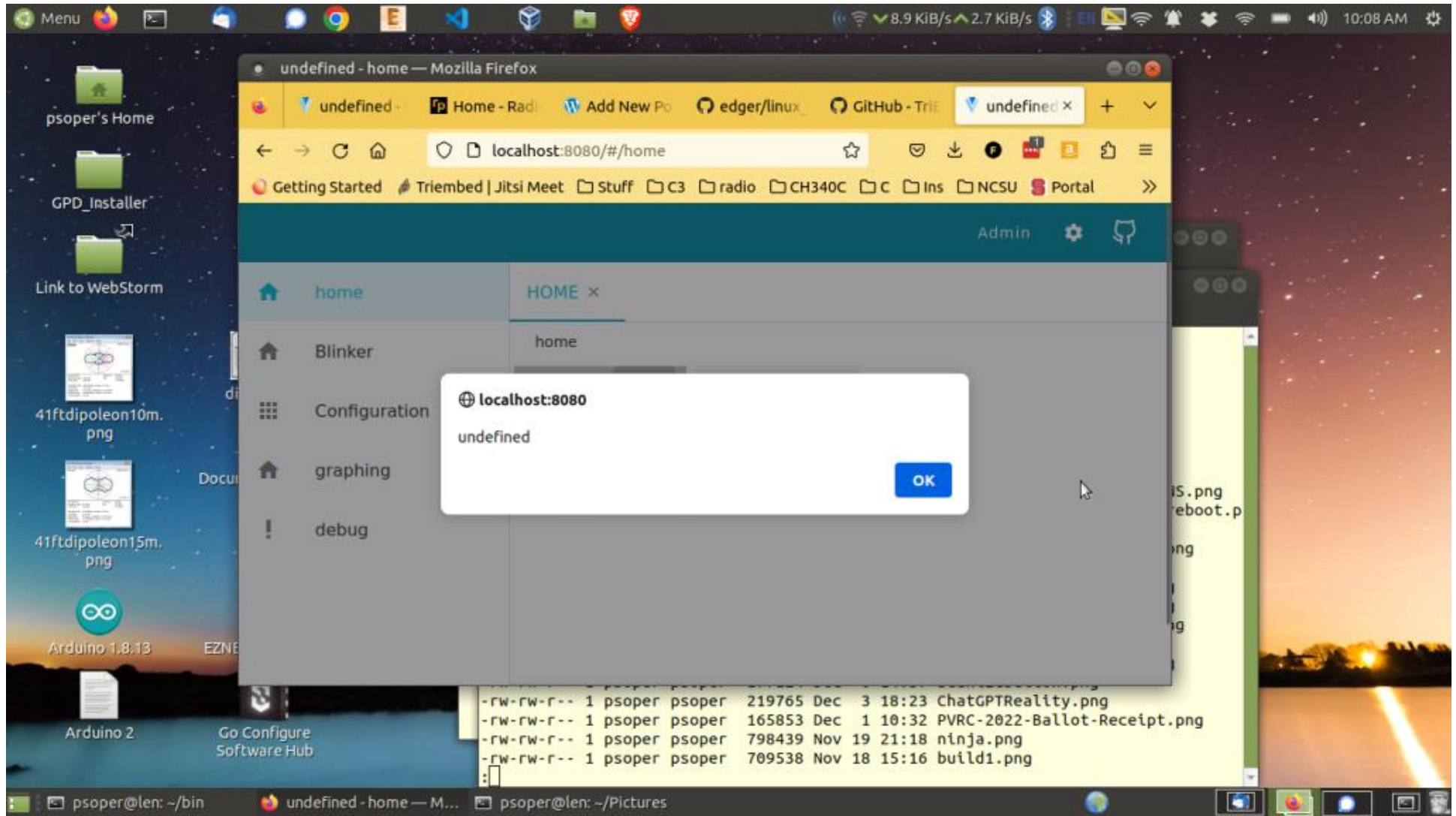
A few steps are needed to prepare the system. First left click “ADMIN” in the upper right of the screen to log out:

v.15.02.2023



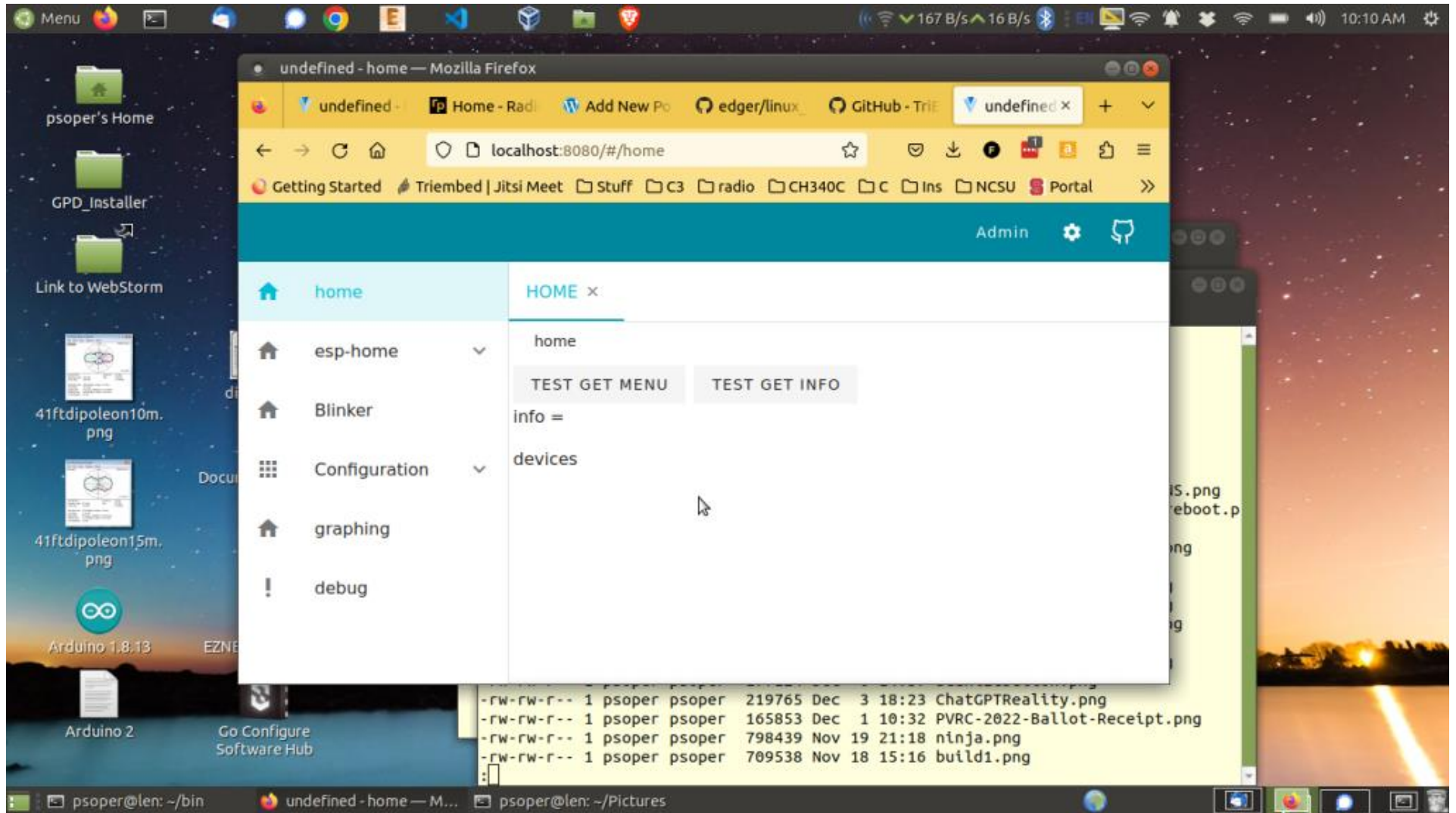
Left click "LOG IN". Now left click "TEST GET MENU" in the middle left:

v.15.02.2023



Left click "OK". Edger is now ready to use:

v.15.02.2023



To test that Edger properly sees the dev board's I2C bus connection to the Renesas FPGA left click "esp-home" on the left of the screen, then "i2c", then "scan". Then left click the "SCANI2C" button:

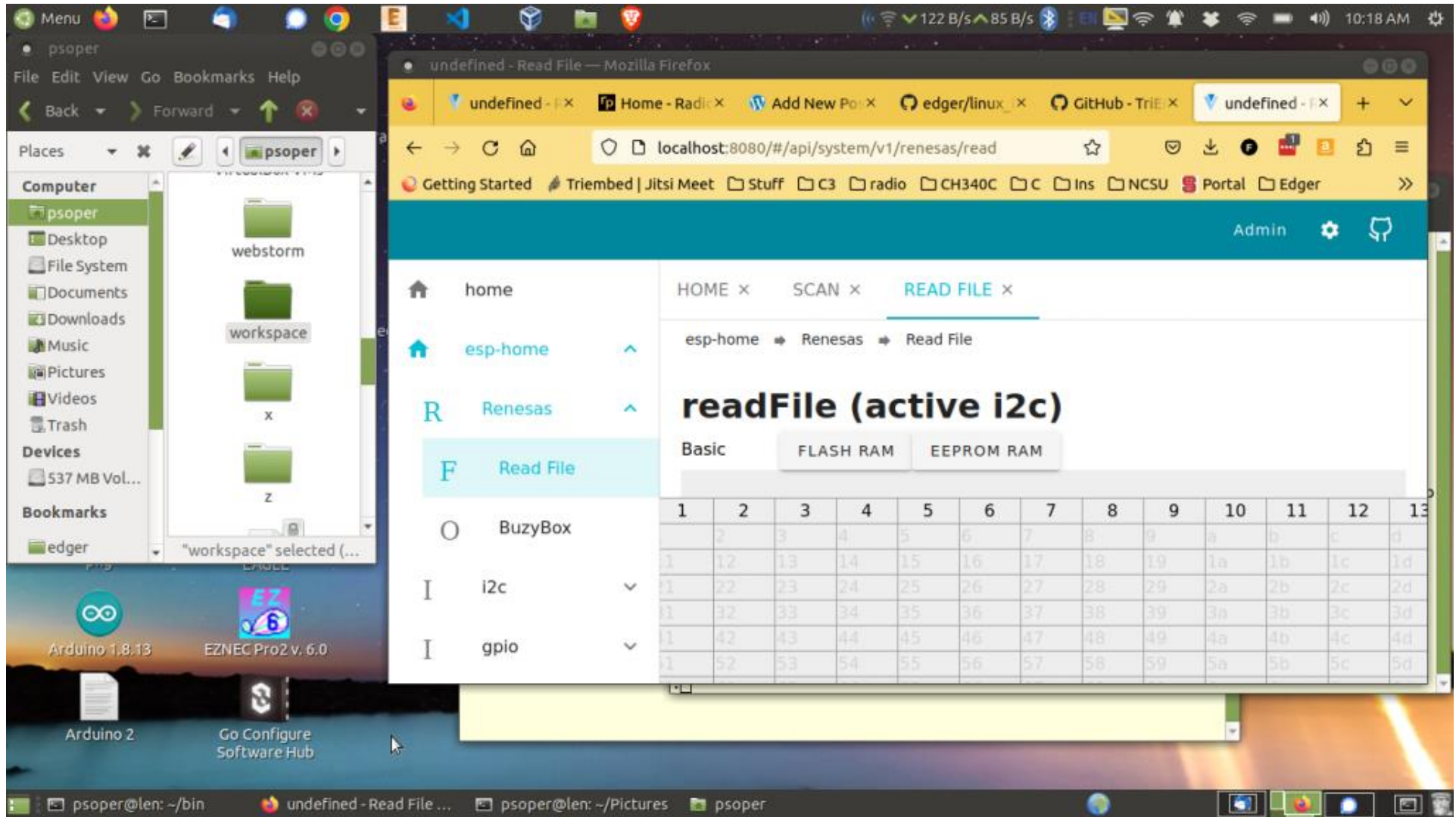
v.15.02.2023

The screenshot shows a Linux desktop with a dark theme. The desktop background is a sunset over a body of water. On the left sidebar, there are several icons: a folder named 'psoper's Home', a folder named 'GPD_Installer', a folder named 'Link to WebStorm', two PNG files named '41ftdipoleon10m.png' and '41ftdipoleon15m.png', an 'Arduino 1.8.13' icon, and an 'Arduino 2' icon. At the bottom of the sidebar, there is a 'Go Configure Software Hub' button. The main window is a Mozilla Firefox browser displaying a web application. The browser's address bar shows 'localhost:8080/#/api/v1/i2c'. The web application has a teal header with 'Admin' and a settings icon. Below the header is a sidebar with a home icon and a list of items: 'esp-home', 'R Renesas', 'I i2c', 'S scan' (highlighted), 'R read', and 'W write'. The main content area shows a 'HOME x SCAN x' tab. Under the 'SCAN x' tab, there is a breadcrumb 'esp-home i2c scan' and a 'SCANI2C' button. Below the button, the text reads: 'Scan (active i2c)' followed by a JSON object: `{"mode": "MASTER", "sda io num": 1, "sda pullup en": 1, "scl io num": 0, "scl pullup en": 1, "master clk speed": 100000}`. Below the JSON is a 5x11 grid of numbers from 0 to 49. The cell containing the number 8 is highlighted in red. At the bottom of the browser window, there is a terminal window showing a file listing:

```
-rw-rw-r-- 1 psoper psoper 219765 Dec 3 18:23 ChatGPTReality.png
-rw-rw-r-- 1 psoper psoper 165853 Dec 1 10:32 PVRC-2022-Ballot-Receipt.png
-rw-rw-r-- 1 psoper psoper 798439 Nov 19 21:18 ninja.png
-rw-rw-r-- 1 psoper psoper 709538 Nov 18 15:16 build1.png
```

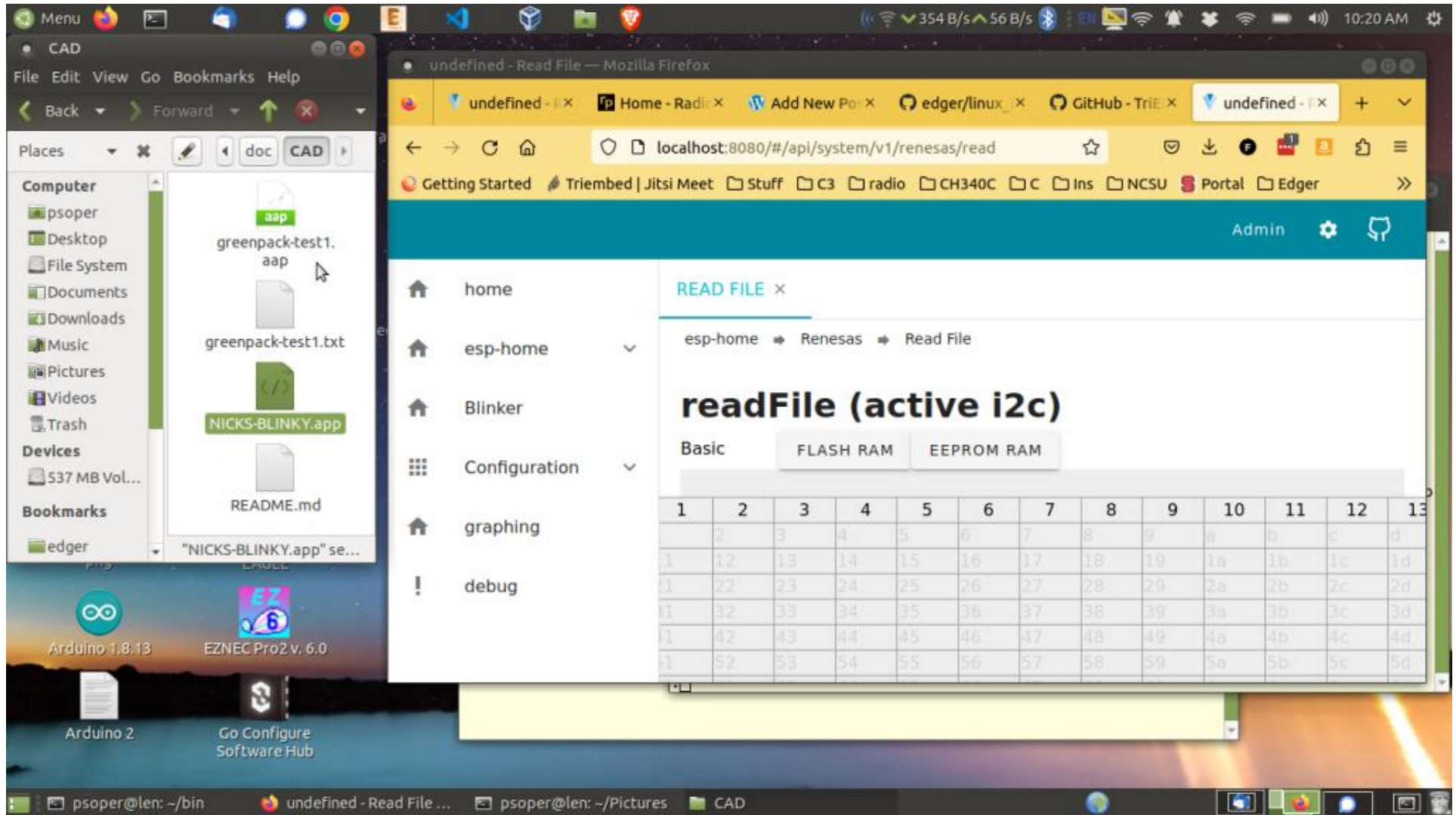
To run the “Blinky” FPGA application left click “Renesas” and then “Read File” and then left click your file manager icon that looks like a file folder to bring up a file chooser dialog:

v.15.02.2023



Navigate from your home directory to subdirectory “workspace”, then “esp32”, then “edger”, then “doc”, then “CAD”:

v.15.02.2023



NOTE that your file manager window will only show a “BLINKY.app” file”. Left click on this file, hold the left mouse button down while dragging the file into the spreadsheet area of the browser page. Release the left mouse button to drop the file into the browser.

v.15.02.2023

Menu

Terminal

File

undefined - Read File — Mozilla Firefox

Start Page

undefined - Read File

Project Edger – Edger Blo

BMP180.pdf

localhost:8080/#/api/system/v1/renesas/read

Getting Started

Triembed | Jitsi Meet

Stuff

C3

radio

CH340C

C

Ins

NCSU

Portal

Edger

Cancer

IDF

Splat

Admin

home

esp-home

R Renesas

F Read File

O BusyBox

I i2c

I gpio

home Blinker

HOME x

READ FILE x

esp-home → Renesas → Read File

readFile (active i2c)

Basic

FLASH RAM

EEPROM RAM

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	F4	F	0	F5	F	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	FC	3F	0	0	0	0	0	0	2	0	80	0
5	0	0	8	4	0	0	FE	1	0	E0	B	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
7	2F	2F	8	0	40	40	4	0	0	0	0	0	0	1	0	1
8	A5	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0

Finally, left click the "FLASH RAM" button and notice that the yellow LED of the dev board is now flashing, confirming that the FPGA has been programmed and is running, feeding a divided down clock signal into the LED. Note the "FLASH EEPROM RAM" button is not yet supported.

To shut your system down cleanly use the upper right "cog/screen" icon with a left click and select "shut down".

