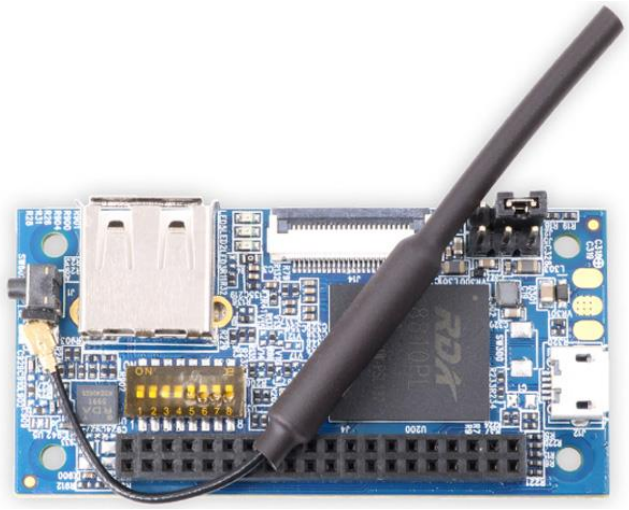

How to make a node of IoT by using OrangePi i96

Juozas Kimtys - How to make a node of IoT by using OrangePi i96 – doc. ver.1.0

Table of contents:

Table of figures:

First impression

Single Board Computer module OrangePi i96 - is it possible to make any useful things based on this module? The module is very small – the size is only 60 mm × 30mm, weight - only 30g. Pitch of 40-pins GPIO connector is **2.0 mm**, not 2.54 mm. Manufacturers page http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-I96.html contains more details. Price at AliExpress was 12€ (2022.02.15), but now (2022.12.07) is about 28€. For comparison - Raspberry Pi 4B 1GB RAM at AliExpress now (2022.11.22) I found price near 140€, Raspberry Pi 3B 1GB RAM at AliExpress now (2022.11.22) I found price about 220€. These prices are already "discounted" by about 50% ("black sale"). Things that make to think that something is wrong with the module OrangePi i96: 1) my module's seller company "Shenzhen Xunlong Software CO.,Limited" now sells only other products from series OrangePi, 2) other sellers at AliExpress do not show any sales history or users comments of product OrangePi i96 (possible they don't have any sales 2022.11.22).

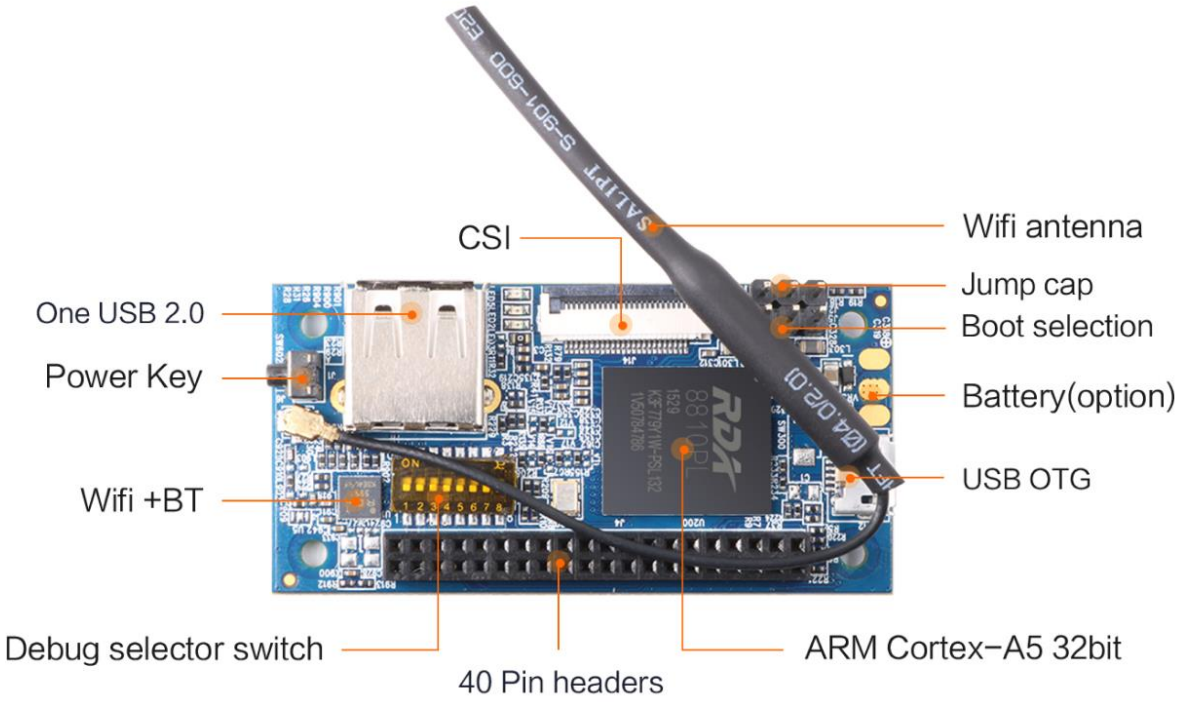

FIGURE 1 - OPTIONS OF ORANGEPI I96 ON PICTURE FROM MANUFACTURER'S SITE

There is no LCD screen connector on the module, but it is mentioned in the Manual. Other product from the same products line and having same processor, is module of type Orange Pi 2G-IoT – some software of this module may be good also for OrangePi i96.

Creating system card

Initially I decided to choose to install Ubuntu Server (not Android, not Debian), because previously many years I used Ubuntu Desktop on many my computers (vitual and physical).

But this my selection was wrong - when my successful investigation appears at the step to check how GPIO pins works, I found, that it is impossible to install WiringPi or WiringOP, because I did not find any source stating to be compatible with OrangePi i96. One dream to overcome this problem was to use Node.js module "onoff", because there was no information about requirement from "onoff" to have WiringPi installed. But module "onoff" rejects installation, because Python3 on Ubuntu 16.04 is too old version - 3.5.x, but requirement is to be not less than 3.6.

At this not very happy step I have found very good page about how to make Debian working on module OrangePi i96 and other useful advices: https://jamesachambers.com/orange-pi-i96-getting-started-guide/

and Debian card image from the same author:

https://github.com/TheRemote/Legendary-OrangePi-i96/releases/ (Debian Bullseye Image v1.36 is current in 2022.11.23)

After downloading to Ubuntu Desktop computer and extracting, I have the file Legendary_OrangePi_i96_debian_bullseye_server_v1.36.img. Inserting USB adapter with inserted 16GB uSD card to computer USB port (not to uSD or to SD slot - if your Ubuntu Desktop runs on VMware virtual machine player) and checking does the system shows up the card. With help of the program Ubuntu Disks - removing all partitions from the card. In Ubuntu files explorer making right mouse click on the previously extracted system disk image file. Selecting option "Open With Disk Image Writer". Writing system image to the card. Note: it is required Ubuntu Desktop machine administrator's password (it is not available on "Try Ubuntu and later install" variant) .

Link below is just for information where are placed other possible disk images for OrangePi i96 at manufacturers server (https://drive.google.com/file/d/1skNNWlSgk3h2GJmjPVWuWQPiGx8-WWM2)

During first run the system (Debian) will resize disk partition (my card is 16GB) to get all available space of the card:

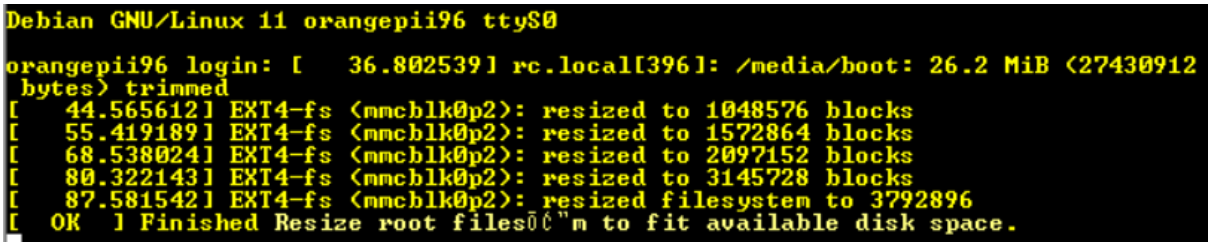

FIGURE 2 - AT FIRST RUN THE SYSTEM (DEBIAN) WILL RESIZE DISK PARTITION.

Serial console

Serial console is mandatory to check does the system works and allows to make initial configuration. In the Manual I didn't find requirements for voltage of serial adapter. From the other side, some forum user in the Internet alerted to not exceed allowed voltage (3V3 or 1V8 ?) to this module's serial input (RX line). For this reason, I made connection of module's TX line to my

oscilloscope, and I found, that voltage is about 3 volts (during startup period of my Ubuntu Server system on uSD card).

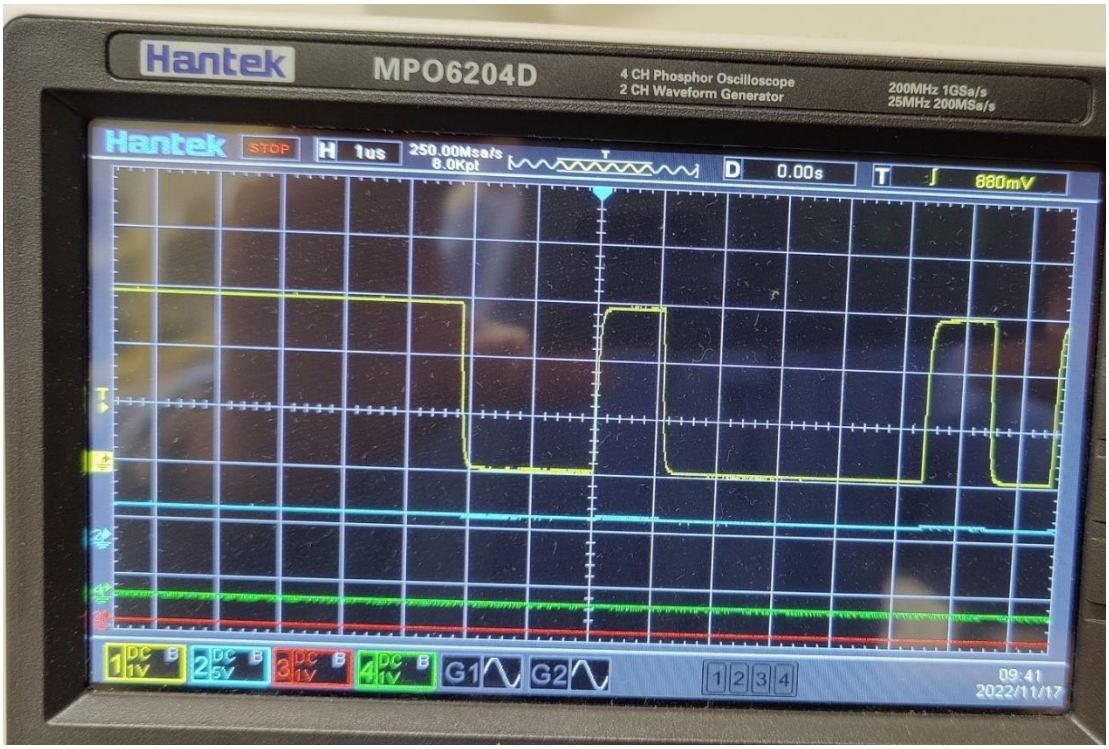

FIGURE 3 - SIGNAL ON TX PIN OF THE MODULE DURING INITIALIZATION OF THE SYSTEM

After, I made modernization of my adapter "USB-to-Serial RX TX " to be "3V3 compliant". Connecting my serial adapter to my Windows computer, checking port number in Device Manager and running the program Realterm and configuring 4 things: speed "**921600**", type of view "**ANSI**", port number, count of rows to watch - 50. Power on the module by using any phone charger having micro-USB connector (5V). Below is the picture of partial dialog in Realterm console window. It works! Typing username orangepi and password orangepi into the console window to login to my Server.

```
Debian GNU/Linux 11 orangepii96 ttyS0

orangepii96 login: orangepi
Password:
Linux orangepii96 3.10.62-rel5.0.2-legendary-v1.36 #27 PREEMPT Mon Oct 24 17:31:
34 MDT 2022 armv7l
**********************************************
Welcome to orangepii96 - Legendary
Please configure your orangepii96 with:
sudo orangepi-config
**********************************************

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 24 08:47:33 EET 2022 from 192.168.43.114 on pts/1
orangepi@orangepii96:~$ [   81.194859] rc.local[433]: /: 12.8 GiB (13749575680 b
ytes) trimmed
[   82.490674] rc.local[488]: /media/boot: 26.2 MiB (27430912 bytes) trimmed
```

FIGURE 4 - DEBIAN CONSOLE FIRST DIALOG.

Note: This screenshot was taken after making all required settings to the system and shows history of last connection from ssh client (it is my windows desktop computer).

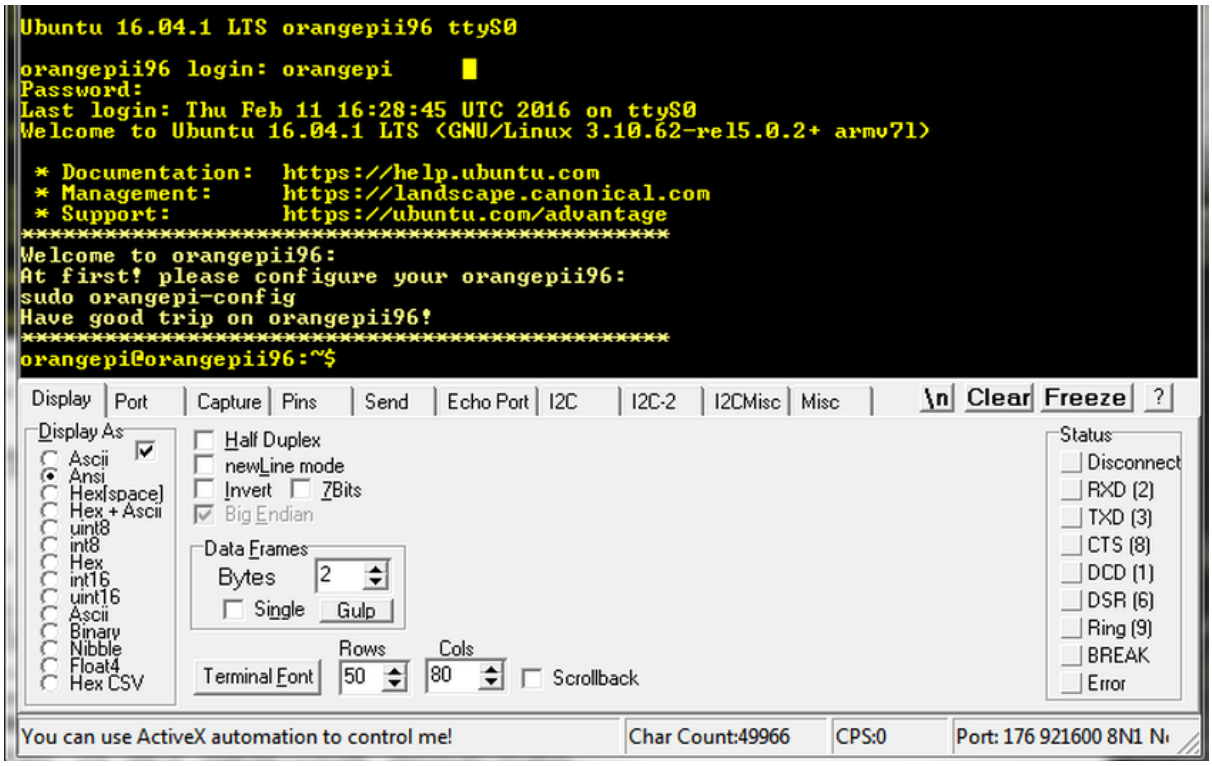

FIGURE 5 - UBUNTU CONSOLE DIALOG AND SETTINGS 'DISPLAY' OF THE PROGRAM REALTERM

Note: This screenshot was taken before making any settings to the system.

Removing alsa

This step is required only for Ubuntu variant. In Debian variant, possible that this step was already included by author of Debian system image.

History because it is required to remove alsa: After making first system setup (setup of WIFI) and cold restart, my server freezes at some point of startup during system initialization. Search on Internet allows me to understand, that I am not alone with similar problems. Some users were unable to start system in any case after reboot, but my case was "the problem after reboot after making first step of system setup". One of suggestions was to remove alsa. One wrote that there is absence of any sound devices on the module, so it will not be a problem to live without alsa.

After reading these suggestions, I created new Ubuntu card (because impossible to use existing - the system freezes after reboot) and running the command:

sudo apt-get purge --auto-remove alsa-utils

This helps!

Configuring Wi-Fi

My module will connect to Wi-Fi and to Internet by using my tablet computer (Android, SIM card, mobile data allowed). In settings of Android finding the function Wi-Fi hotspot, entering some

useful SSID name and strong password, allowing this hotspot. In Serial terminal window - login to my Ubuntu/Debian Server (user orangepi, password orangepi) and running the command:

sudo orangepi-config

Pay attention to the exact name of the command - the Manual of OrangePi i96 contains wrong name of this command. Right command text is from initial dialog in the console window of this server during initial system dialog.

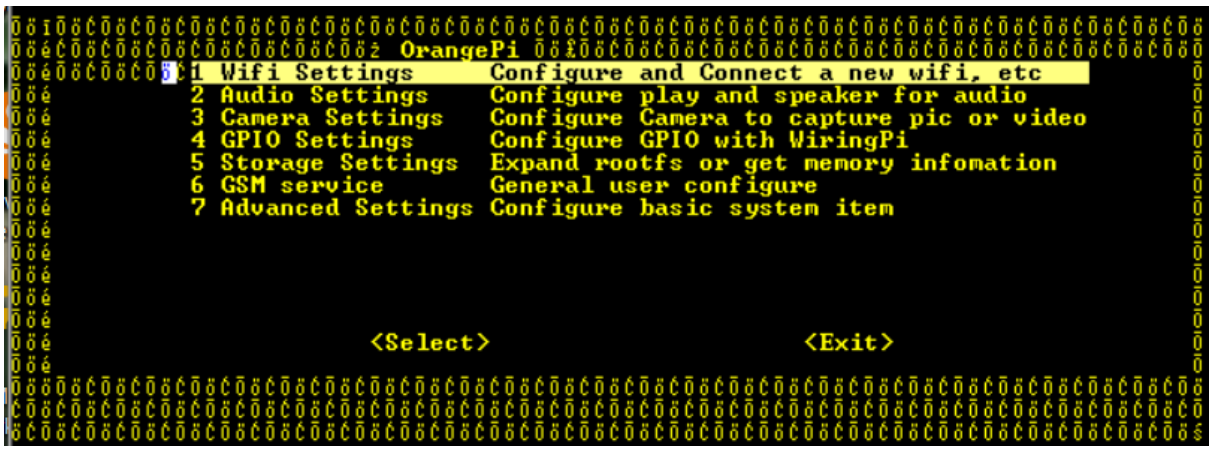

FIGURE 6 – DEBIAN CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG

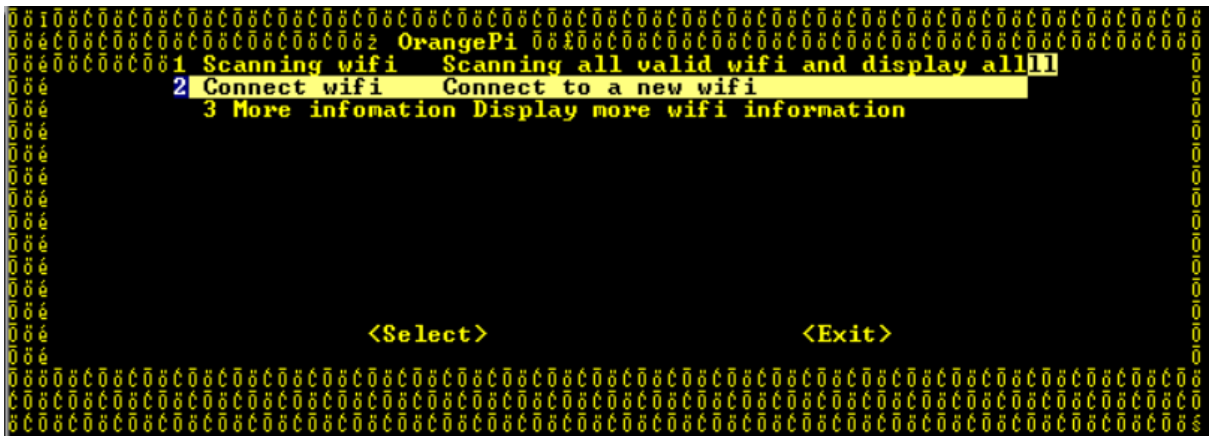

FIGURE 7 - SELECTIONS TO CONFIGURE WI-FI

FIGURE 8 - CONSOLE DIALOG TO ENTER WI-FI AP NAME

Pictures above contain information about console dialog during system configuration initiated by running sudo orangepi-config. Using "arrow up" or "arrow down" or "tab" keys to

navigate through controls. Selecting "Connect Wi-Fi" and in newly opened dialog entering WI-FI SSID and password. Doing reboot of the system by using the command:

sudo reboot

After reboot, login to the server and running a command

ifconfig (Ubuntu) or *ip addr* (Debian)

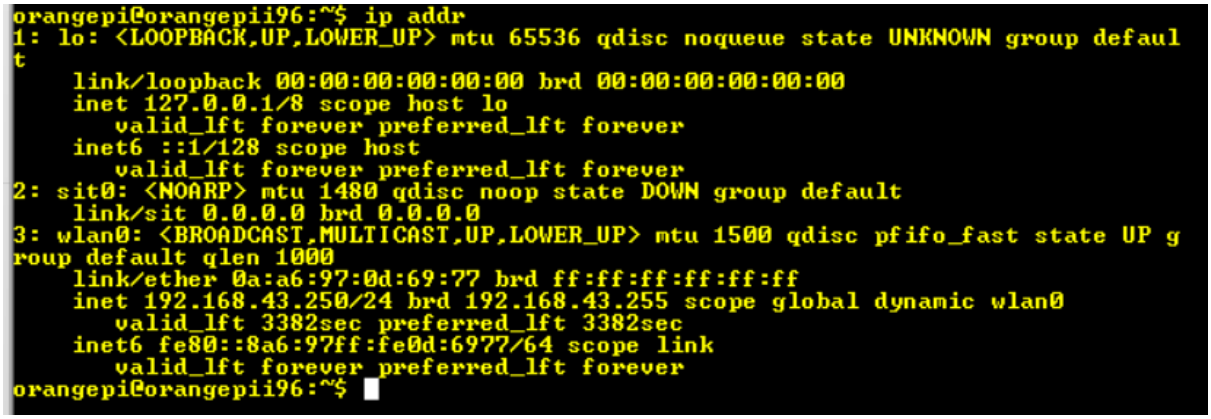

FIGURE 9 - DEBIAN CONSOLE AFTER THE COMMAND 'IP ADDR'

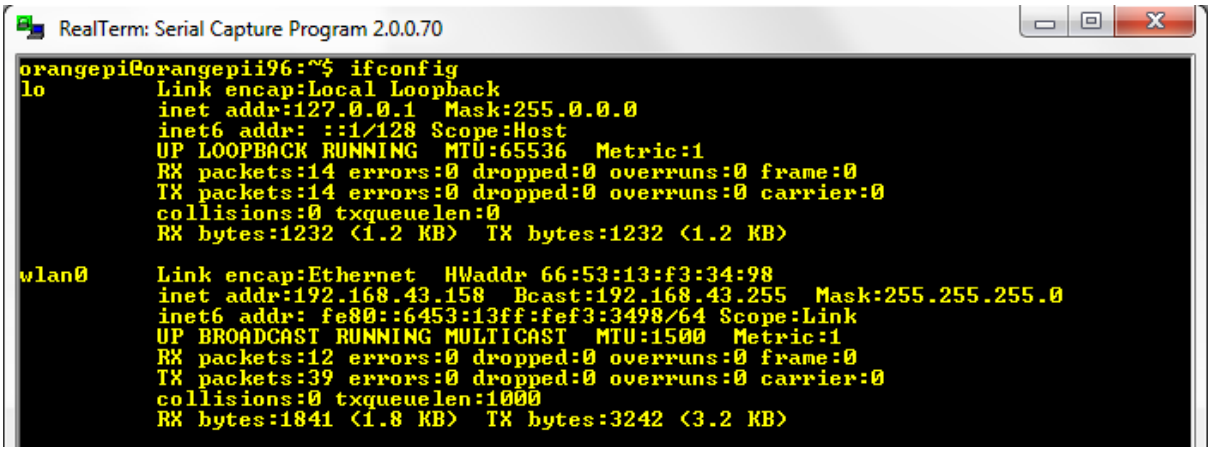

FIGURE 10 - UBUNTU CONSOLE AFTER THE COMMAND 'IFCONFIG'

Pictures above show what IP address of my server is. On my Ubuntu system: IP address will remain the same during a day - if to not to reenable Wi-Fi hotspot or to not cold start my Ubuntu system. But, on my Debian, IP address stays the same for many days independently of power on/off of the module and of the hotspot.

Configuring SSH

In Serial terminal window - login to my server (user orangepi, password orangepi) and running the command:

sudo orangepi-config

Pay attention to the exact name of the command - the Manual of OrangePi i96 contains wrong name of this command. Right command text is from initial dialog in the console window of

this server during initial system dialog. Picture below contain information about console dialog during system configuration initiated by running sudo orangepi-config. Using "arrow up" or "arrow down" or "tab" keys to navigate through controls. Selecting "Advanced Settings" and setting program switch to allow SSH. Also, in Advanced settings - changing host name from default "orangepii96" to custom "orangepii96-juozo1".

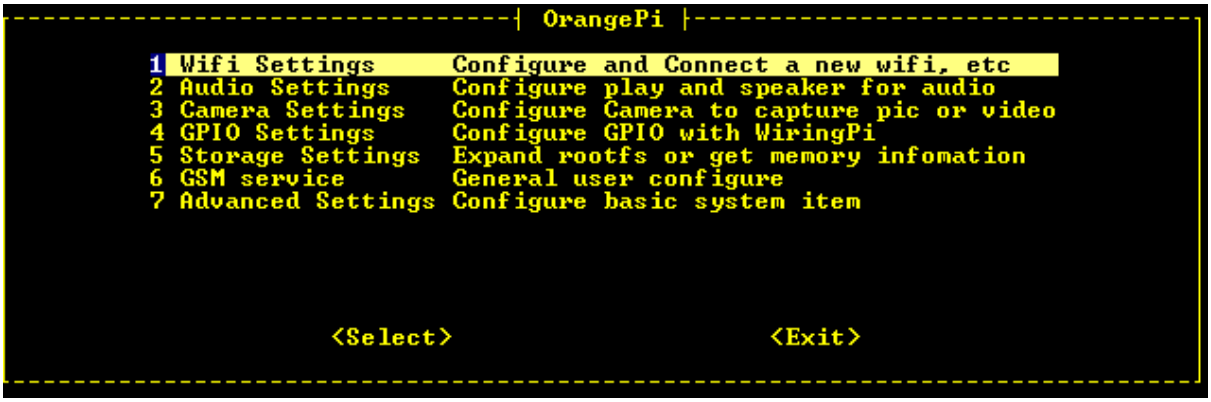

FIGURE 11 - UBUNTU CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG

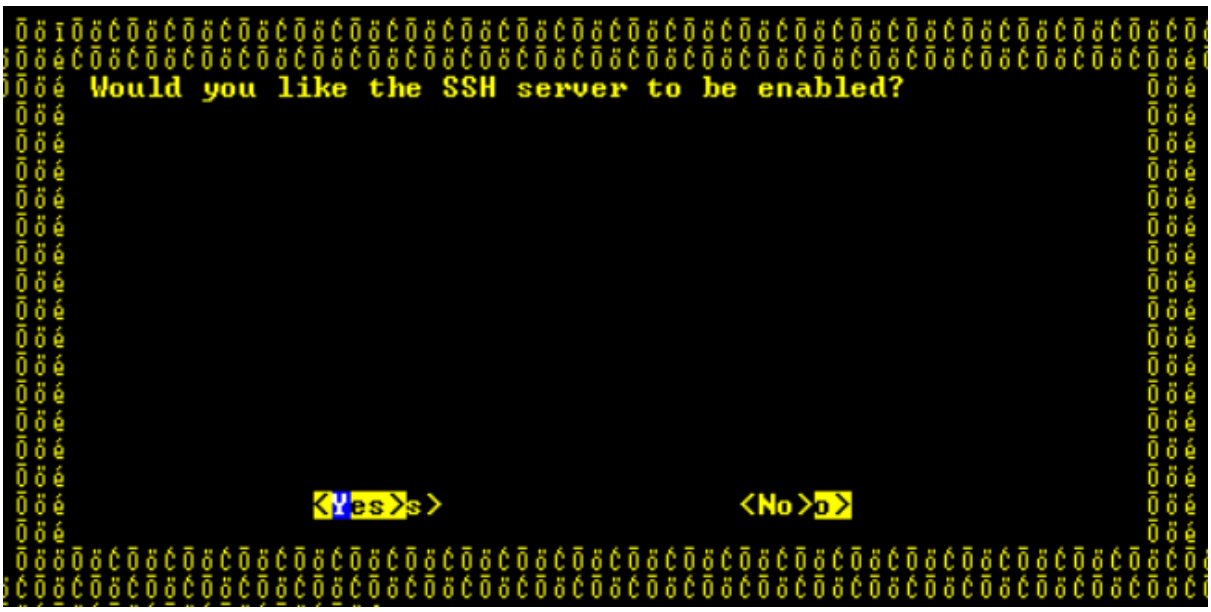

FIGURE 12 - DEBIAN CONSOLE DIALOG ENABLE SSH (UNDER ADVANCED SETTINGS)

Doing cold reboot of the system by using the command:

sudo halt

After disconnecting and reconnecting power, login to my module's system (Ubuntu or Debian) from Windows computer by using PowerShell tool by using the command ssh orangepi@192.168.43.209. Note1: Serial console is still required to be sure about IP address of my system on module. Note2: my Windows computer's Wi-Fi must be connected to the same Wi-Fi hotspot as the module.

Pictures below shows communication session from Windows PowerShell (version 7.3.0).

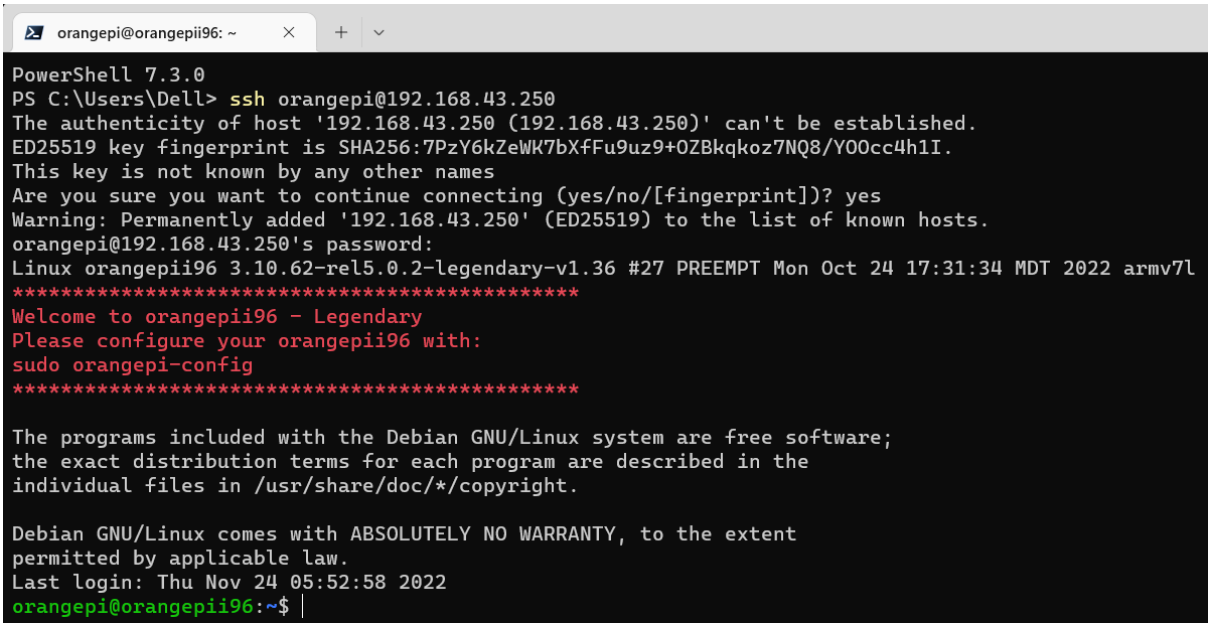

FIGURE 13 – DEBIAN SSH CONSOLE SESSION FROM WINDOWS POWERSHELL

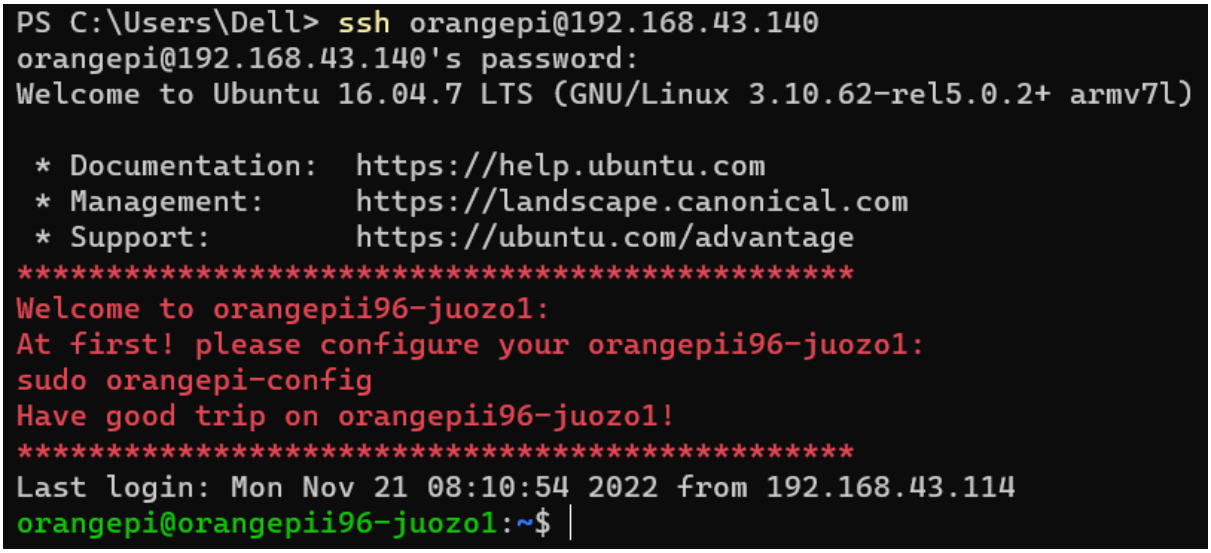

FIGURE 14 – UBUNTU SSH CONSOLE SESSION FROM WINDOWS POWERSHELL

Configuring GPIO

In Serial terminal window - login to my Ubuntu Server (user orangepi, password orangepi) and running the command:

sudo orangepi-config

Pay attention to the exact name of the command - the Manual of OrangePi i96 contains wrong name of this command. Right command text is from initial dialog in the console window of this server during initial system dialog.

Picture below contain information about console dialog during system configuration initiated by running sudo orangepi-config. Using "arrow up" or "arrow down" or "tab" keys to navigate through controls. Selecting "GPIO Settings" and finding additional selections.

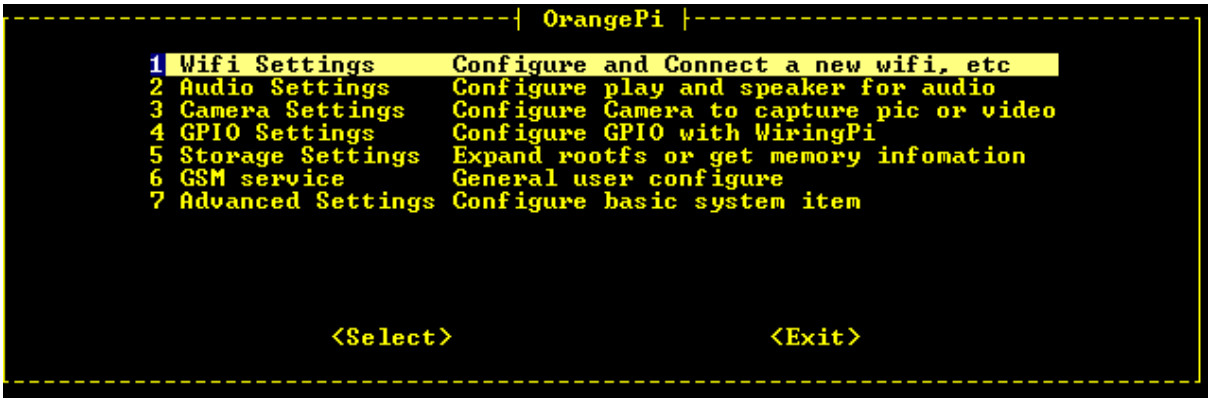

FIGURE 15 - UBUNTU CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG

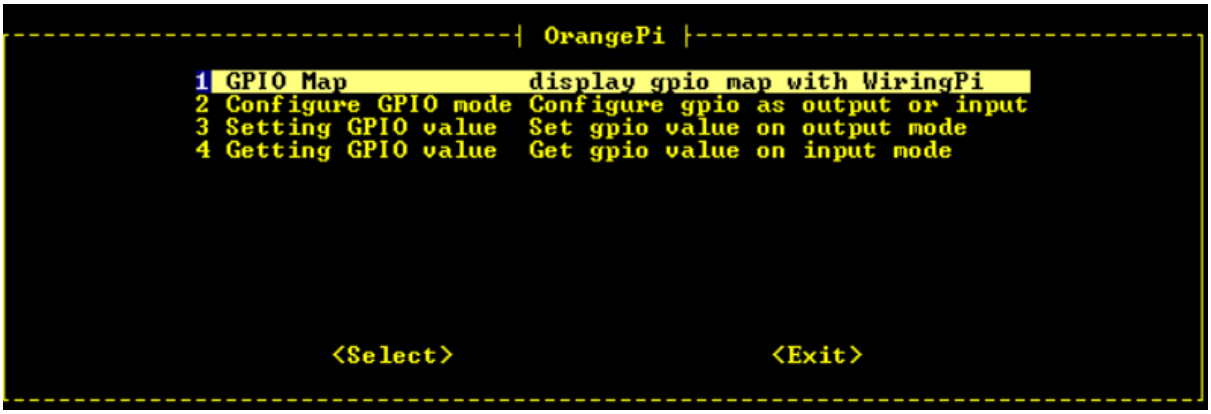

FIGURE 16 - UBUNTU CONSOLE DIALOG GPIO SETTINGS

```
orangepi@orangepii96:~$ sudo gpio readall
[sudo] password for orangepi:
 +-----+-----+----------+------+---+-Orange Pi I96+---+---+------+----------+-----+-----+
 | RDA | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | RDA |
 +-----+-----+----------+------+---+----++----+---+------+----------+-----+-----+
 |     |     | GND      |      |   |  1 || 2  |   |      | GND      |     |     |
 |  40 |  40 | UART2.CTS |  OUT | 0 |  3 || 4  |   |      | PWR_BTN_N |     |     |
 | 104 | 104 | UART2.TX |  ALT | 1 |  5 || 6  |   |      | RST_BTN_N |     |     |
 | 103 | 103 | UART2.RX |  ALT | 1 |  7 || 8  | 0 |  ALT | SPI2.CLK | 2   | 2   |
 |  41 |  41 | UART2.RTS |  OUT | 0 |  9 || 10 | 0 |   IN | SPI2.DI  | 4   | 4   |
 |  14 |  14 | UART1.TX |  ALT | 1 | 11 || 12 | 0 |  ALT | SPI2.CS1 | 6   | 6   |
 | 102 | 102 | UART1.RX |  ALT | 1 | 13 || 14 | 0 |  ALT | SPI2.DO  | 3   | 3   |
 |   0 |   0 | I2C2.SCL |  ALT | 1 | 15 || 16 | 0 |  ALT | I2S_LRCK | 10  | 10  |
 |   1 |   1 | I2C2.SDA |  ALT | 1 | 17 || 18 | 0 |  ALT | I2S.BCK  | 9   | 9   |
 |  38 |  38 | I2C3.SCL |  ALT | 1 | 19 || 20 | 0 |  ALT | I2S.DO   | 13  | 13  |
 |  39 |  39 | I2C3.SDA |  ALT | 1 | 21 || 22 | 0 |  ALT | I2S.DI   | 11  | 11  |
 |  15 |  15 | G.A A15  |  ALT | 0 | 23 || 24 | 1 |  ALT | G.B A20  | 20  | 20  |
 |  56 |  56 | G.C B24  |   IN | 1 | 25 || 26 | 0 |  ALT | G.D D02  | 66  | 66  |
 |  67 |  67 | G.E D03  |  ALT | 0 | 27 || 28 | 1 |  ALT | G.F A22  | 22  | 22  |
 |  30 |  30 | G.G A30  |  ALT | 0 | 29 || 30 | 0 |  ALT | G.H A29  | 29  | 29  |
 |  28 |  28 | G.I A28  |  ALT | 0 | 31 || 32 | 0 |  ALT | G.J 27   | 27  | 27  |
 |  26 |  26 | G.K A26  |  ALT | 0 | 33 || 34 | 0 |  ALT | G.L A25  | 25  | 25  |
 |     |     | V_PAD    |      |   | 35 || 36 |   |      | SYS_DCIN |     |     |
 |     |     | VDD_IN   |      |   | 37 || 38 |   |      | SYS_DCIN |     |     |
 |     |     | GND      |      |   | 39 || 40 |   |      | GND      |     |     |
 +-----+-----+----------+------+---+----++----+---+------+----------+-----+-----+
 | RDA | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | RDA |
 +-----+-----+----------+------+---+-Orange Pi I96+---+------+----------+-----+-----+
```

FIGURE 17 - DEFAULT GPIO MAP AFTER THE COMMAND SUDO GPIO READALL

It must be possible to change direction (IN or OUT) and state (1 or 0) of any pin by using GPIO settings dialog. But only one pin I found working on my Debian system (**pin "56"**) – because almost all pins by default are set to do ALT function. **But none of GPIO functions work on Ubuntu because something wrong is on my Ubuntu image with existing GPIO libraries, trying to install wiringPi or wiringOP appears is not possible on this my Ubuntu, because too old version of Python on this Ubuntu and it is not supported module type (processor type) by xunlong's wiringOP**.

Happily, it is a good solution with Debian – there is a good system disk image prepared by https://jamesachambers.com/orange-pi-i96-getting-started-guide/

Here is additional; good link about this theme: https://discuss.96boards.org/t/resources-for-the-i96-orangepi/11444

And code to change initial configuration of GPIO is here: https://wiki.pbeirne.com/patb/i96/src/master/gpio_fixup.py from the site: https://wiki.pbeirne.com/patb/i96

Here I will need the command "wget", which is absent on Debian by default. Use these actions:

$ sudo apt-get update

Install the newest versions of all packages currently installed#

$ sudo apt-get upgrade

$ sudo apt install wget

Starting to get gpio_fixup.py:

cd /usr/local/bin

wget http://wiki.pbeirne.com/patb/i96/raw/master/gpio_fixup.py

```
orangepi@orangepii96:/$ cd usr
orangepi@orangepii96:/usr$ cd local
orangepi@orangepii96:/usr/local$ cd bin
orangepi@orangepii96:/usr/local/bin$ sudo wget http://wiki.pbeirne.com/patb/i96/raw/master/gpio_fixup.py
--2022-11-29 07:13:47--  http://wiki.pbeirne.com/patb/i96/raw/master/gpio_fixup.py
Resolving wiki.pbeirne.com (wiki.pbeirne.com)... 158.51.124.249
Connecting to wiki.pbeirne.com (wiki.pbeirne.com)|158.51.124.249|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wiki.pbeirne.com/patb/i96/raw/master/gpio_fixup.py [following]
--2022-11-29 07:13:48--  https://wiki.pbeirne.com/patb/i96/raw/master/gpio_fixup.py
Connecting to wiki.pbeirne.com (wiki.pbeirne.com)|158.51.124.249|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: 'gpio_fixup.py'

gpio_fixup.py                   [ <=>                                       ]   2.35K  --.-KB/s    in 0s

2022-11-29 07:13:50 (8.12 MB/s) - 'gpio_fixup.py' saved [2405]
```

FIGURE 18 - CONSOLE DIALOG GETTING GPIO_FIXUP.PY

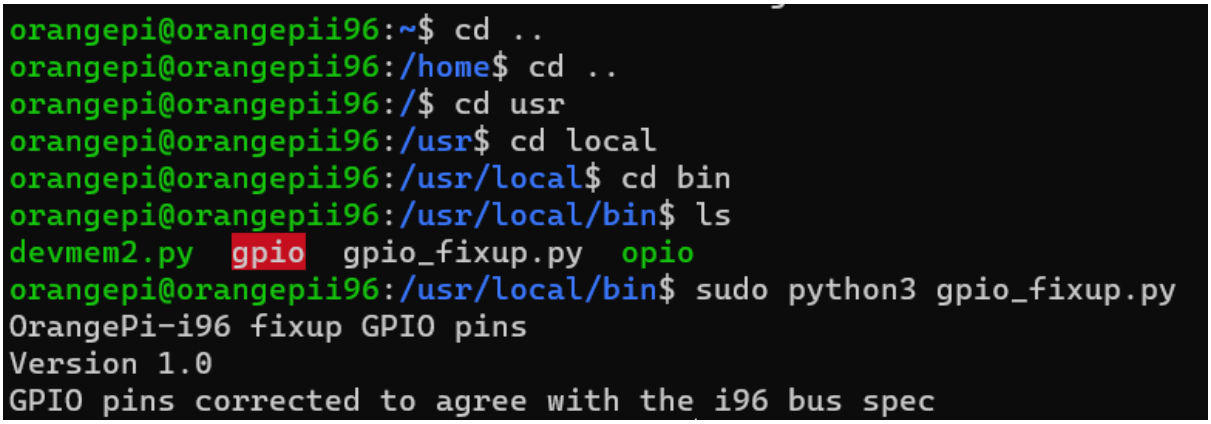

FIGURE 19 - CONSOLE DIALOG EXECUTING SUDO PYTHON3 GPIO_FIXUP.PY

Next – I must find how to run the command “Python3 gpio_fixup.py” during system boot.

Proposal of the author of the script gpio_fixup.py:

> # add a line to /etc/rc.local to execute this at startup:
>
> *sudo sed -i "/^exit 0$/i\/usr/local/bin/gpio_fixup.py" /etc/rc.local*
>
> Modifying permissions:
>
> *cd /usr/local/bin*
>
> *sudo chmod +x* gpio_fixup.py

Now GPIO map is correct:

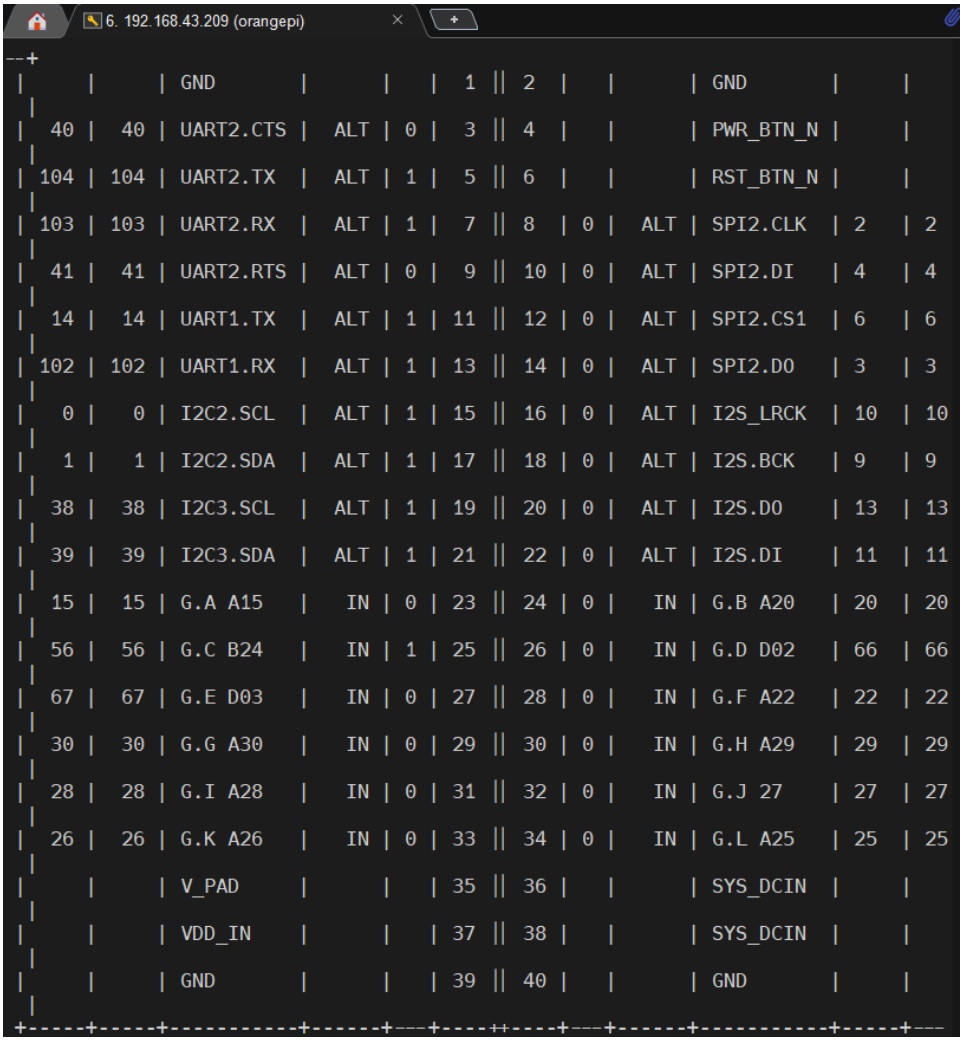

FIGURE 20 - GPIO MAP ON MOBAXTERM CONSOLE DIALOG AFTER EXECUTING GPIO_FIXUP.PY

Installing Node.js

Latest Node.js version now is "18".

But I have found, that the Node.js module onoff supports Node.js versions 10, 12, 14, 15 and 16. So, I am going to install version "16".

Using some advices from the page: https://www.makersupplies.sg/blogs/tutorials/how-to-install-node-js-and-npm-on-the-raspberry-pi

sudo apt-get update

sudo apt-get upgrade

uname -m (get processor type)

Now I know that my processor's type code is "**armv7l** " (note: L (small cap) is at the end of the code, not the digit 1!).

Note: Windows PowerShell (7) allows to use clipboard to copy-paste commands into (and from) SSH session dialog window. So, I am using SSH session on Windows PowerShell starting from below steps. Also, noting that the command *sudo -i* (to become root user) works on this system.

So, my file to download is: https://nodejs.org/download/release/v16.18.1/node-v16.18.1-linux-armv7l.tar.xz

Need additional archiving tools, because "xz":

sudo apt install xz-utils

sudo apt-get install wget (need only if Debian (if not installed on some previous step), not if Ubuntu)

Running the command to download the archive:

wget https://nodejs.org/download/release/v16.18.1/node-v16.18.1-linux-armv7l.tar.xz

Extracting the archive with the command:

tar -xf node-v16.18.1-linux-armv7l.tar.xz

Going to directory of extracted files:

cd node-v16.18.1-linux-armv7l/

Making these files accessible from any place:

*sudo cp -R * /usr/local/*

Checking node and npm version:

node -v

npm -v

Seeing that node version is 16.18.1 and npm version is 8.19.2.

Deleting now unneeded archive:

cd ..

sudo rm node-v16.18.1-linux-armv7l.tar.xz

Setting my local time zone

On Debian I saw that system automatically gets time from network. It is good, but I must correct time zone settings:

date

Thu Nov 24 06:36:52 UTC 2022

sudo timedatectl list-timezones

Find my zone from the list.

Change time zone:

sudo timedatectl set-timezone Europe/Vilnius

Now my time is:

Thu Nov 24 08:42:56 EET 2022

Installing Node-RED

It is required to have Node.js - I already installed it in section before this.

Using advice from https://agrinode.github.io/docs/install_nodered_orangepi/

sudo npm install -g --unsafe-perm node-red

For Debian: the system, after node-red installation, proposed to update npm and I did this:

sudo npm install -g npm@9.2.0

Now check how my node-red works:

node-red

Response in Console window:

Welcome to Node-RED

8 Dec 10:34:56 - [info] Node-RED version: v3.0.2

8 Dec 10:34:56 - [info] Node.js version: v16.18.1

8 Dec 10:34:56 - [info] Linux 3.10.62-rel5.0.2-legendary-v1.36 arm LE

8 Dec 10:35:02 - [info] Loading palette nodes

8 Dec 10:35:10 - [info] Settings file : /root/.node-red/settings.js

8 Dec 10:35:10 - [info] Context store : 'default' [module=memory]

8 Dec 10:35:10 - [info] User directory : /root/.node-red

8 Dec 10:35:10 - [warn] Projects disabled : editorTheme.projects.enabled=false

8 Dec 10:35:10 - [info] Flows file   : /root/.node-red/flows.json

8 Dec 10:35:10 - [info] Creating new flow file

8 Dec 10:35:10 - [warn]

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials file will not be recoverable, you will have to delete it and re-enter your credentials.

You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change.

8 Dec 10:35:10 - [info] Server now running at http://127.0.0.1:1880/

8 Dec 10:35:10 - [warn] Encrypted credentials not found

8 Dec 10:35:10 - [info] Starting flows

8 Dec 10:35:10 - [info] Started flows

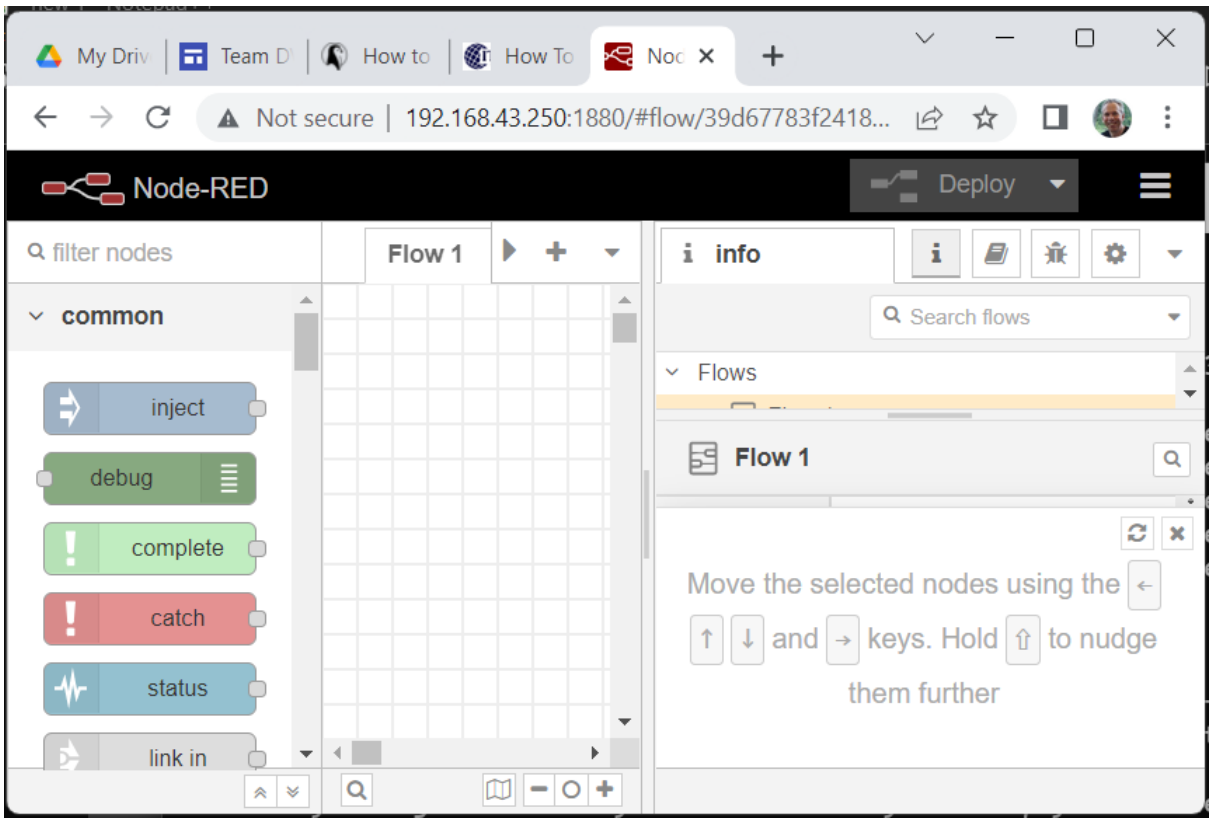

FIGURE 21 - MY SERVER'S NODE-RED WEB PAGE ON BROWSER RUNNING ON MY WINDOWS

Open and edit files with MobaXterm

I want to open and edit on my Windows computer text and code files located inside my server. Checked how works the program Sublime Text 4 and found this program working and useful. But Sublime Text 4 is not free to use to anyone. After I found other program, which is free for home using and contains impressive additional function to remote browse files and also many other good functions.

Download page is this: https://mobaxterm.mobatek.net/download-home-edition.html

Using of this program is very easy - just install this program on Windows, run and enter SSH connection data - no need any additional steps to do in server installation (in opposite to Sublime Text 4).

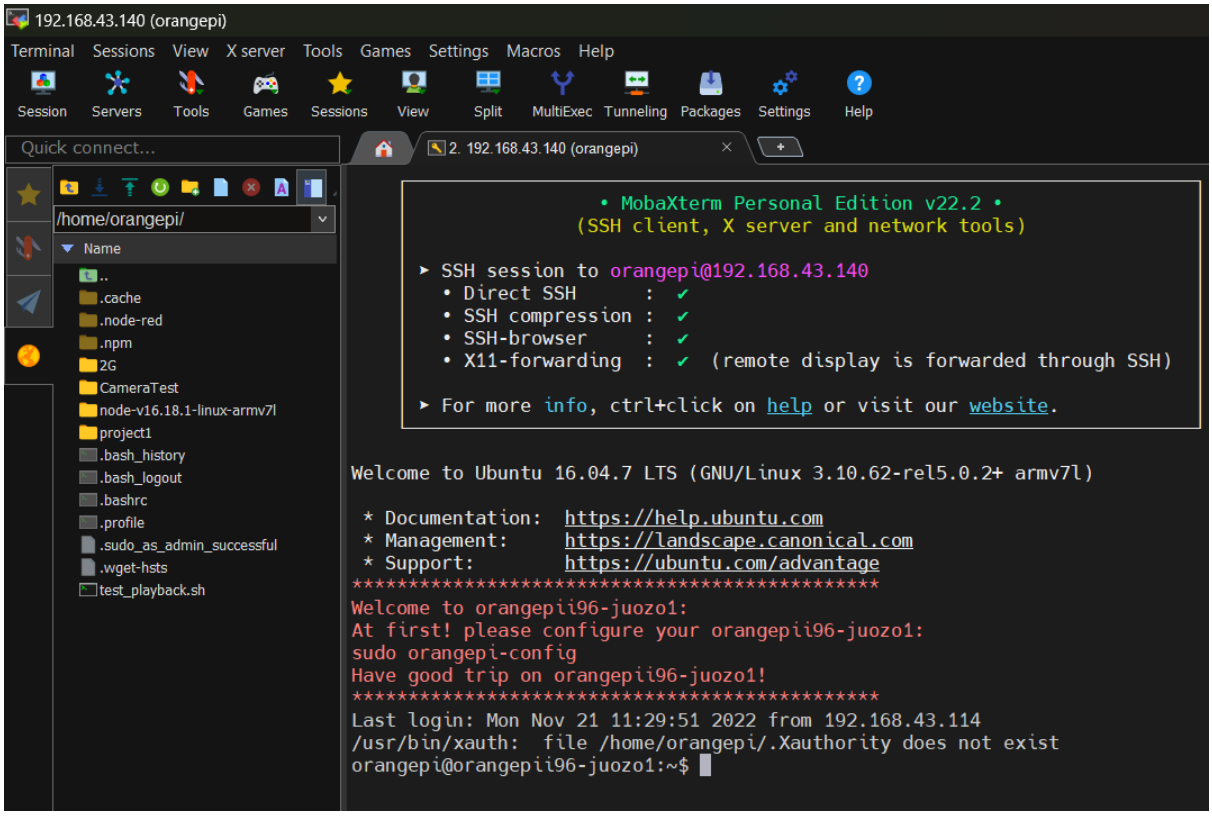

FIGURE 22 - MOBAXTERM PROGRAM WINDOW AFTER CONNECTION TO MY SERVER

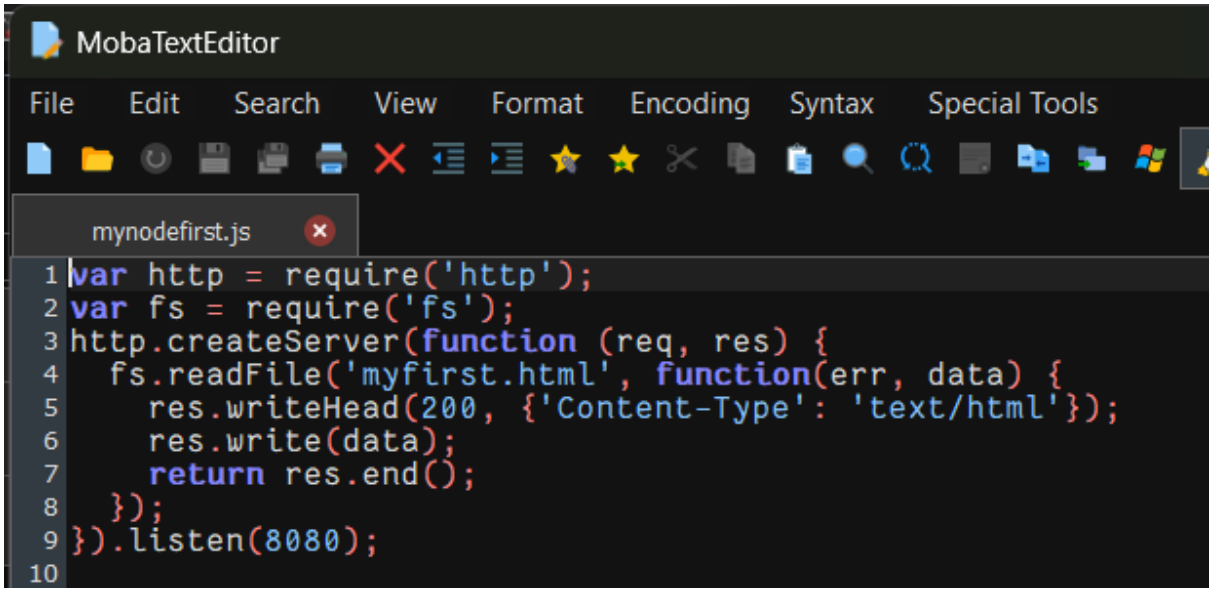

FIGURE 23 - EDITING MY FIRST NODE.JS PROGRAM ON MOBATEXTEDITOR FOR THIS DEVICE

```
orangepi@orangepii96-juozo1:~$ ls
2G  CameraTest  node-v16.18.1-linux-armv7l  project1  test_playback.sh
orangepi@orangepii96-juozo1:~$ cd project1
orangepi@orangepii96-juozo1:~/project1$ node mynodefirst.js
```

FIGURE 24 - RUNNING MY FIRST NODE.JS PROGRAM FOR THIS DEVICE

FIGURE 25 - WEB PAGE EXAMPLE GENERATED BY NODE.JS ON MY ORANGEPI I96 SERVER

Blink LED (Node.js)

In order to toggle on-off GPIO pins, I need the module "onoff" .
https://www.npmjs.com/package/onoff

Many other GPIO libraries and methods are well described here (Comparing node.js GPIO implementations): https://gist.github.com/jperkin/e1f0ce996c83ccf2bca9

npm install onoff

npm install spi-device – we will need this later

On my Debian system the module 'onoff' I have installed successfully because Python version is 3.9.2.

But on Ubuntu 16.04 installation of this module fails, because too old version of Python3 (it is 3.5.2 version but required >Python3.6).

To use "onoff" on Ubuntu 16.04, one must install newer Python3.6. But it seems impossible task to do on Ubuntu 16.04

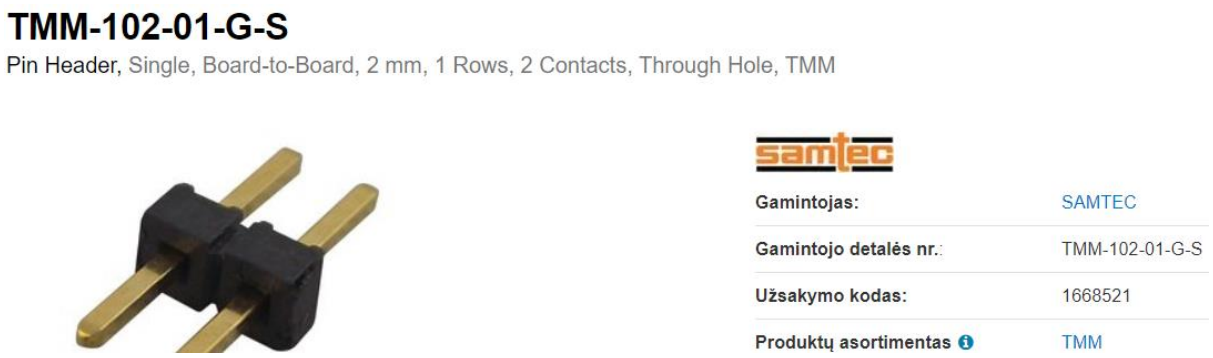

FIGURE 26 - FINDING CONNECTORS HAVING PITCH 2.0MM (SCREENSHOT FROM FARNELL)

FIGURE 27 - FINDING CONNECTORS HAVING PITCH 2.0MM (CJK4202K/2 FROM RCL.LT)

Now I'm using Debian. Connecting oscilloscope probes to 40-pins GPIO connector. Now I'm using only GPIO_B24 (pin25 by schematic of the module or pin56 by WiringPi) because other pins

initially are configured (by Debian firmware) to serve as ALT functions and are not accessible by using nor Node.js library “onoff”, nor the program orangepi-config. But, after running gpio_fix.py (during boot – as I configured previously), all pins must be accessible.

Creating some code. Initial code for this testing was taken from the: https://www.w3schools.com/nodejs/nodejs_raspberrypi_blinking_led.asp

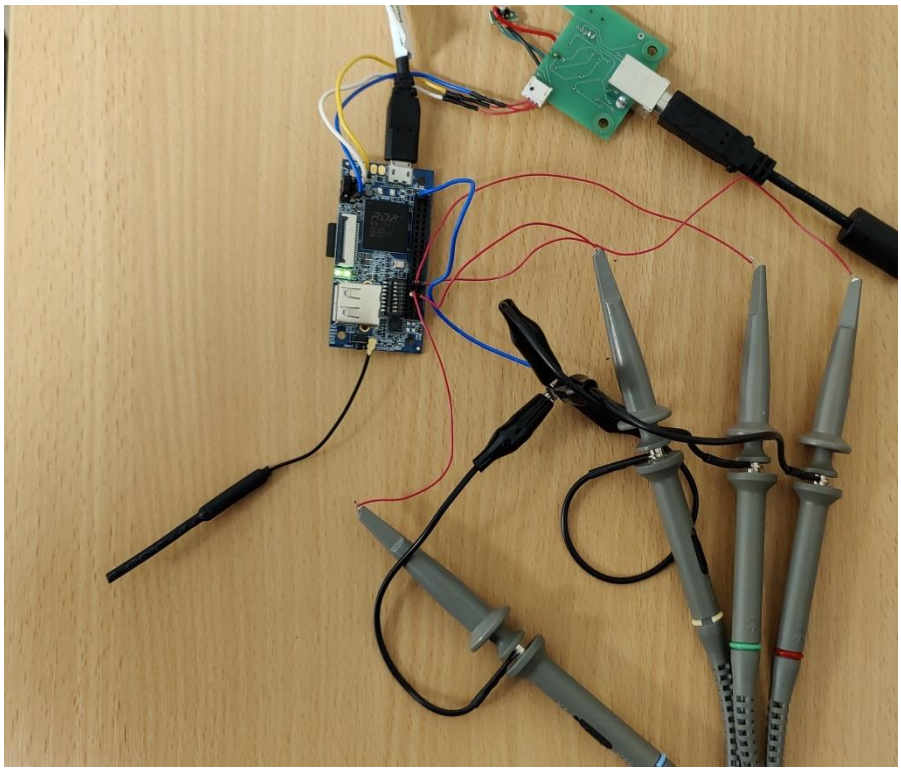

FIGURE 28 - CONNECTING OSCILLOSCOPE PROBES TO SOME GPIO PINS

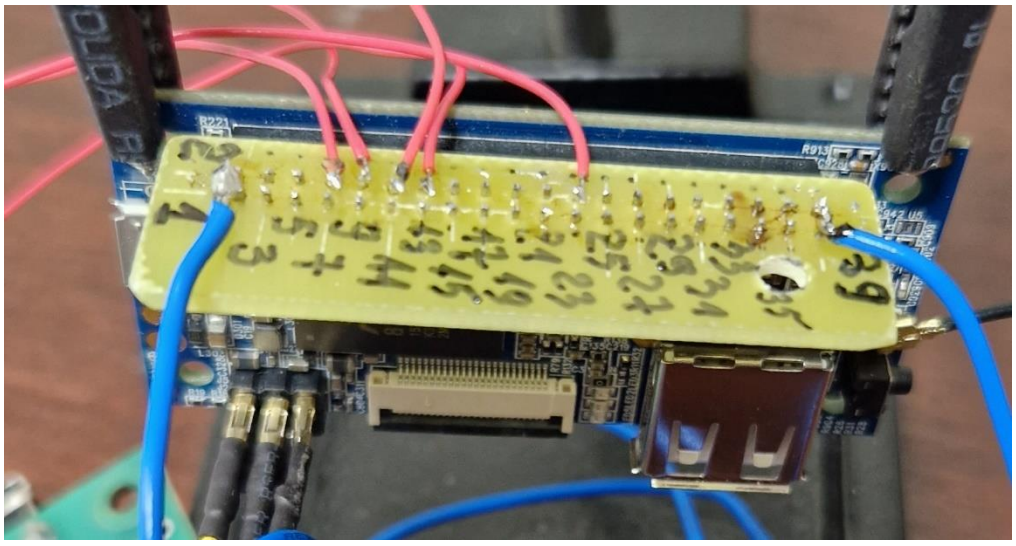

FIGURE 29 - PROTOTYPE BOARD TO TEST GPIO

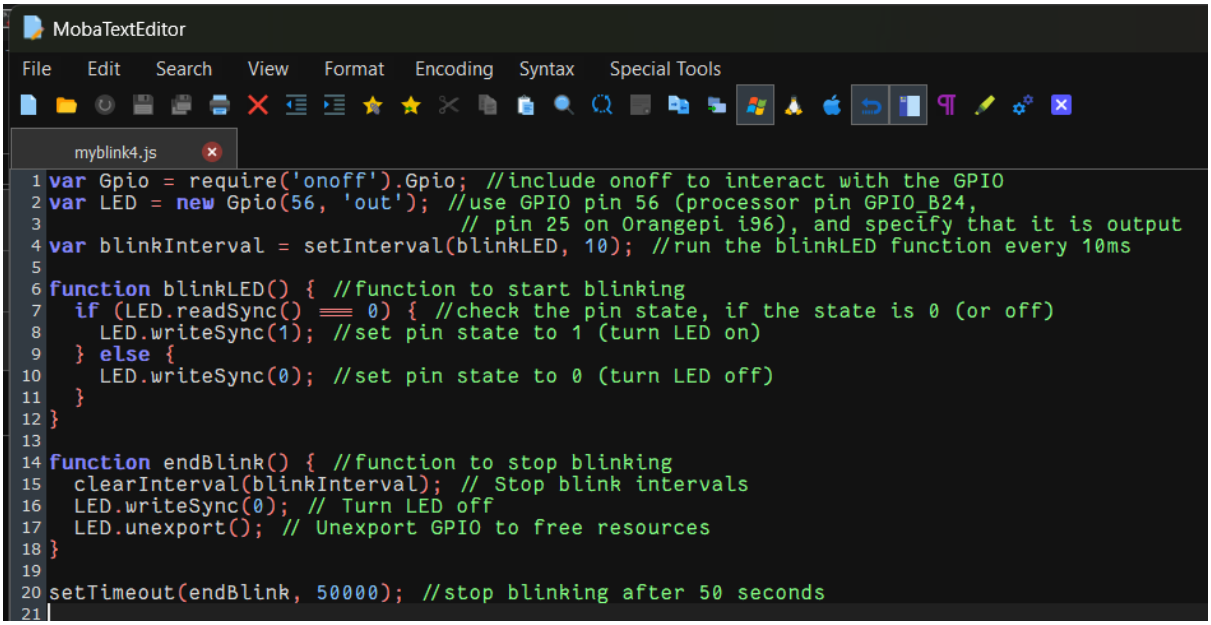

FIGURE 30 - MAKING NODE.JS CODE ON MOBAXTERM MOBATEXTEDITOR

From any of my terminal session (Serial console, Windows PowerShell or MobaXterm):

sudo node myblink4.js

I need to do "sudo" because there are required permissions to access pins. It is possible to stop execution earlier than code ends - by using CTRL+Z

Using SPI ADC (Node.js)

To make measurement with SPI ADC, I need the module "spi-device" https://www.npmjs.com/package/spi-device/v/3.1.2 ,

SPI works, but I found impossible to change SPI speed from “20.000.000Hz” to any other value. The module "spi-device" functions “open” and “setOptions” really do not change **anything** from default initial, because it is platform specific and do not really were implemented for this board. My ADC need to change speed to some 100000Hz or less, I’m stopping my attempts to make measurements with SPI ADC.

My code:

```
/*
Test ADC of type MCP3208 on Debian 11 by using Node.js
Juozas Kmtys
2022.12.14
using examples from:
https://www.npmjs.com/package/spi-device/v/3.1.2

*/

var Gpio = require('onoff').Gpio; //include onoff to interact with the GPIO
const spi = require('spi-device');
var measurementInterval1 = setInterval(myFunc_make_measurement1, 200);

// I will not use special pin !SPI2_CS, because the library 'spi-device' do
not have posibility to switch to other CS pin (t0 use more than one SPI device
on same bus)

var myGpio_CS_MCP3208 = new Gpio(15, 'out'); //use GPIO pin 15, and specify
that it is output
myGpio_CS_MCP3208.writeSync(1); //set pin state to 1 (deselect ADC)
var myGpio_CS_EEPROM = new Gpio(20, 'out'); //use GPIO pin 20, and specify
that it is output
myGpio_CS_EEPROM.writeSync(1); //set pin state to 1 (deselect EEPROM)

const intChannelCodeInitial = 0x80;
var intChannelCodeCurrent = intChannelCodeInitial;
const intCountOfChannels = 8;
var blnFlagSpiOptionsNeedToBeSet = true;

console.log('Starting spi-device test');

myGpio_CS_MCP3208.writeSync(0); //set pin state to 0 (select ADC)

//-------------------------------------
```

```
function myFunc_make_measurement1() {

    myGpio_CS_MCP3208.write(1, (err) => { //set pin state to 1 (deselect ADC)
        if (err) throw err;
        myGpio_CS_MCP3208.write(0, (err) => { //reset pin state to 0 (select
ADC)
            if (err) throw err;
// The MCP3008 is on bus 1 and it's device 0
const mcp3008 = spi.open(1, 0, err => {
    // An SPI message is an array of one or more read+write transfers
    const message = [{
      sendBuffer: Buffer.from([0x01, intChannelCodeCurrent, 0x00]), // Sent to
read channel 5
      receiveBuffer: Buffer.alloc(3),              // Raw data read from
channel 5
      byteLength: 3,
      speedHz: 20000 // Use a low bus speed to get a good reading from the -
but this setting really does not work on the board OrangePi i96 on Debian
(Juozas Kimtys), I will try to set it soon separately
    }];

    if (err) throw err;

    if (blnFlagSpiOptionsNeedToBeSet)
    {
    mcp3008.getOptions( (err, options) => {
        if (err) throw err;
        console.log('Initial options:', options);
        const myNewOptions = [{
            maxSpeedHz: 20000 // initial value was '20000000', but this
setting also really does not work on the board OrangePi i96 on Debian
          }]

          mcp3008.setOptions(myNewOptions, (err) => {
            if (err) throw err;
            blnFlagSpiOptionsNeedToBeSet = false;
          })
    });
    } //   if (blnFlagSpiOptionsNeedToBeSet)

    mcp3008.transfer(message, (err, message) => {
      if (err) throw err;

      // Convert raw value from sensor to celcius and log to console
      const rawValue = ((message[0].receiveBuffer[1] & 0x0F) << 8) +
        message[0].receiveBuffer[2];
      const voltage = rawValue * 2.5 / 4095;
```

```
      console.log(intChannelCodeCurrent, ' : ', rawValue);
      intChannelCodeCurrent += 1;
      if (intChannelCodeCurrent == (intChannelCodeInitial +
intCountOfChannels))
      {
        intChannelCodeCurrent = intChannelCodeInitial;
      }
    }); // mcp3008.transfer(message
  }); // const mcp3008 = spi.open(1
        }); // myGpio_CS_MCP3208.write(0
    }); // myGpio_CS_MCP3208.write(1
} // function myFunc_measurement1

//--------------------------------------

//--------------------------------------

function myFunc_stop_measurement1() {
  clearInterval(measurementInterval1); // Stop blink intervals
  //LED1.write(0); // Turn LED off
  //LED1.unexport(); // Unexport GPIO to free resources
}

//--------------------------------------

setTimeout(myFunc_stop_measurement1, 2000000); //stop measurements after 2000
seconds
```