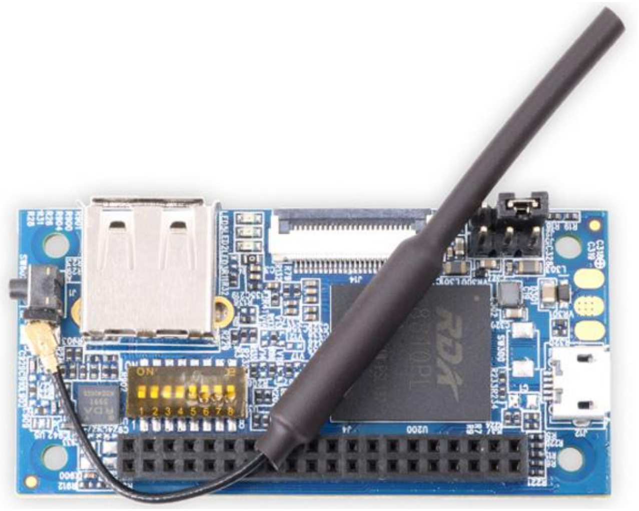


How to make a node of IoT by using OrangePi i96

Juozas Kimtys - How to make a node of IoT by using OrangePi i96 – doc. ver.2.0.1

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).

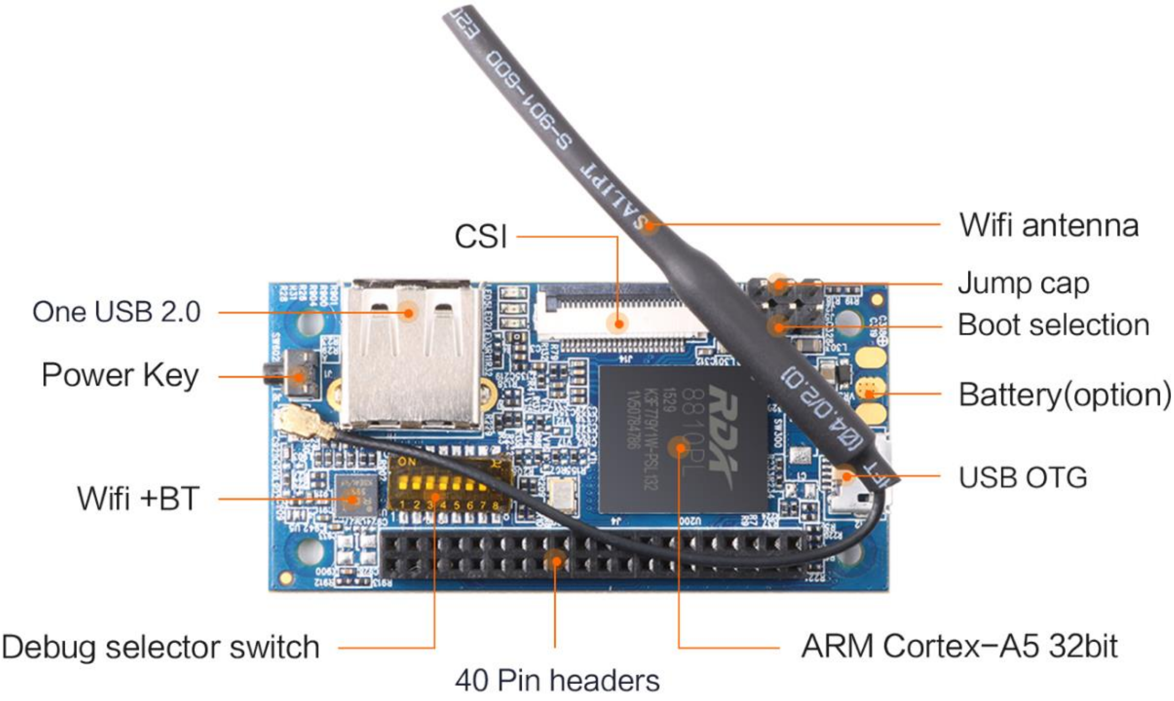


FIGURE 1 - OPTIONS OF ORANGEPI I96 ON PICTURE FROM MANUFACTURER'S SITE

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 Debian system card

Below is information link how to make Debian system on module OrangePi i96: https://jamesachambers.com/orange-pi-i96-getting-started-guide/

Link to Debian card image file from the same author:

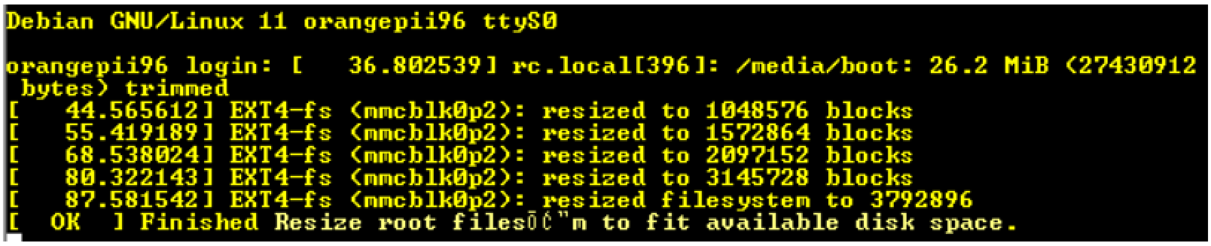
https://github.com/TheRemote/Legendary-OrangePi-i96/releases/ (Debian Bullseye Image v1.36 is current in 2022.11.23)

We are using Ubuntu Desktop inside virtual machine (VMware Player) hosted on Windows.

After downloading and extracting, we 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 Ubuntu Desktop runs on VMware virtual machine player) and checking does the Ubuntu system shows up the card. With help of the program 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 - there are placed other possible disk images (Android, Debian and Ubuntu) for OrangePi i96 on manufacturers server: https://drive.google.com/file/d/1skNNWlSgk3h2GJmjPVWuWQPiGx8-WWM2

During first run the system (Debian) will resize disk partition (our card is 16GB) to get all available space of the card. If later we will decide to make a backup copy of this card (to have all things configured and saved), we will understand that less size card would be better.



```
Debian GNU/Linux 11 orangepii96 ttyS0

orangepii96 login: [   36.802539] rc.local[396]: /media/boot: 26.2 MiB (27430912
 bytes) trimmed
[   44.565612] EXT4-fs (mmcblk0p2): resized to 1048576 blocks
[   55.419189] EXT4-fs (mmcblk0p2): resized to 1572864 blocks
[   68.538024] EXT4-fs (mmcblk0p2): resized to 2097152 blocks
[   80.322143] EXT4-fs (mmcblk0p2): resized to 3145728 blocks
[   87.581542] EXT4-fs (mmcblk0p2): resized filesystem to 3792896
[  OK  ] Finished Resize root filesÕ¢"m to fit available disk space.
```

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 we 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, we can made connection of module's TX line to oscilloscope, and we will find, that voltage is about 3 volts (during startup period of Debian system on uSD card).

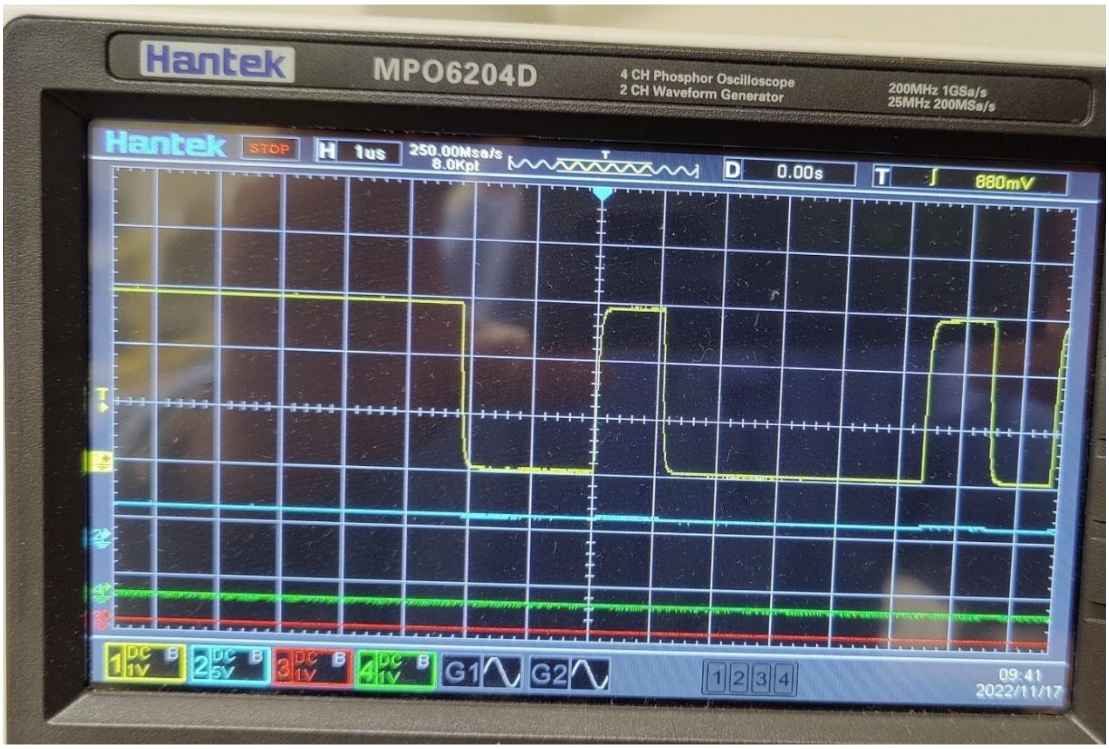
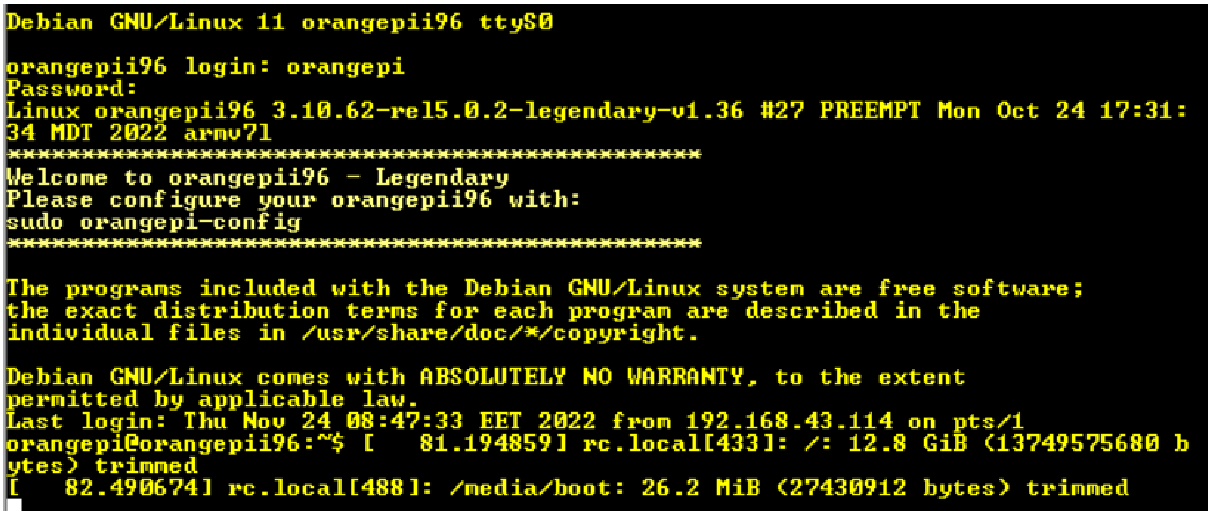


FIGURE 3 - SIGNAL ON TX PIN OF THE MODULE DURING INITIALIZATION OF THE SYSTEM

After, we are making modernization or switching of adapter "USB-to-Serial RX TX " to be "3V3 compliant". Connecting serial adapter to Windows computer, checking port number in Device Manager, 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. Typing username orangepi and password orangepi into the console window to login to our Debian 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.

Removing alsa

This step is required to do manually 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, Ubuntu Server freezes at some point of startup during system initialization. Search on Internet allows to understand, that other users have had similar problems. Some users were unable to start system in any case after reboot, but our 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.

Configuring WiFi

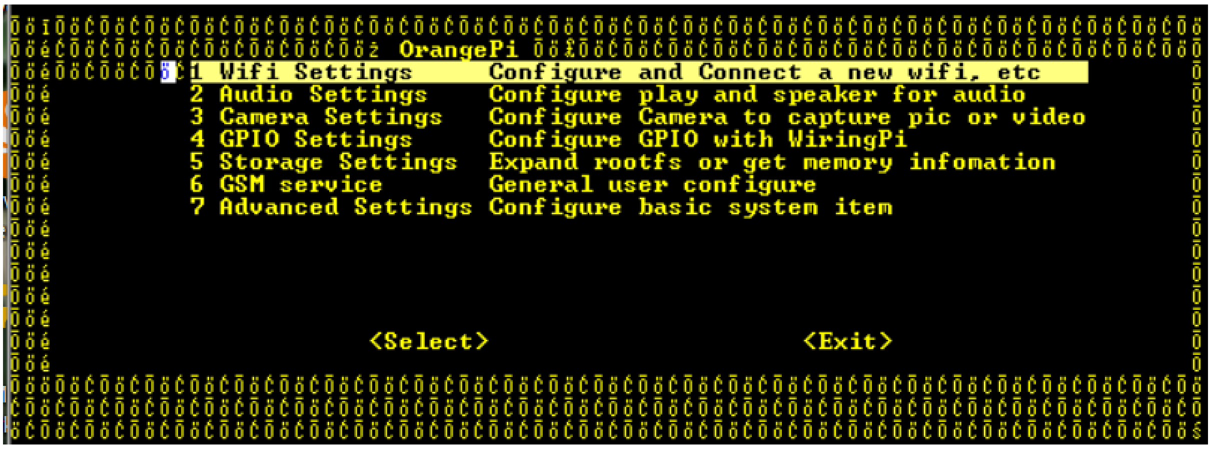
Our server will be connected to Internet by using tablet computer (Android 9, SIM card, mobile data allowed).

In settings of Android finding the function WiFi hotspot, entering some useful SSID name and strong password, allowing this hotspot.

In Windows Serial terminal window - login to our 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.



```
                         OrangePi
1 Wifi Settings       Configure and Connect a new wifi, etc
2 Audio Settings      Configure play and speaker for audio
3 Camera Settings     Configure Camera to capture pic or video
4 GPIO Settings       Configure GPIO with WiringPi
5 Storage Settings    Expand rootfs or get memory infomation
6 GSM service         General user configure
7 Advanced Settings   Configure basic system item



            <Select>                     <Exit>
```

FIGURE 5 – DEBIAN CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG

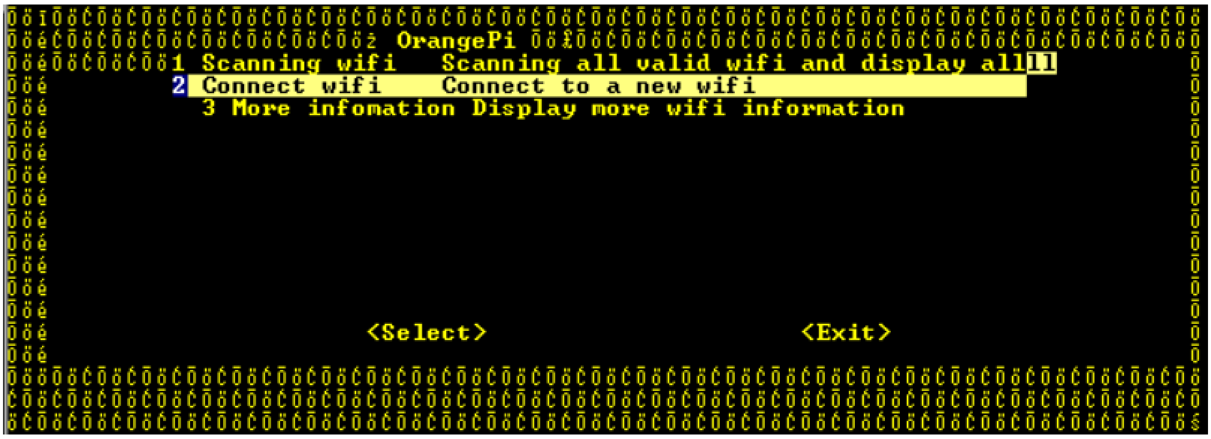

OrangePi
1 Scanning wifi   Scanning all valid wifi and display all
2 Connect wifi   Connect to a new wifi
3 More infomation Display more wifi information
<Select>
<Exit>

FIGURE 6 - SELECTIONS TO CONFIGURE WI-FI

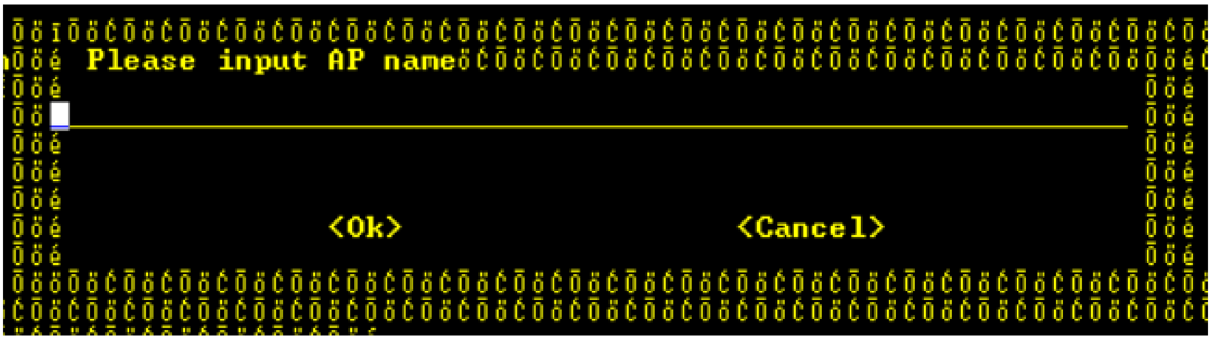


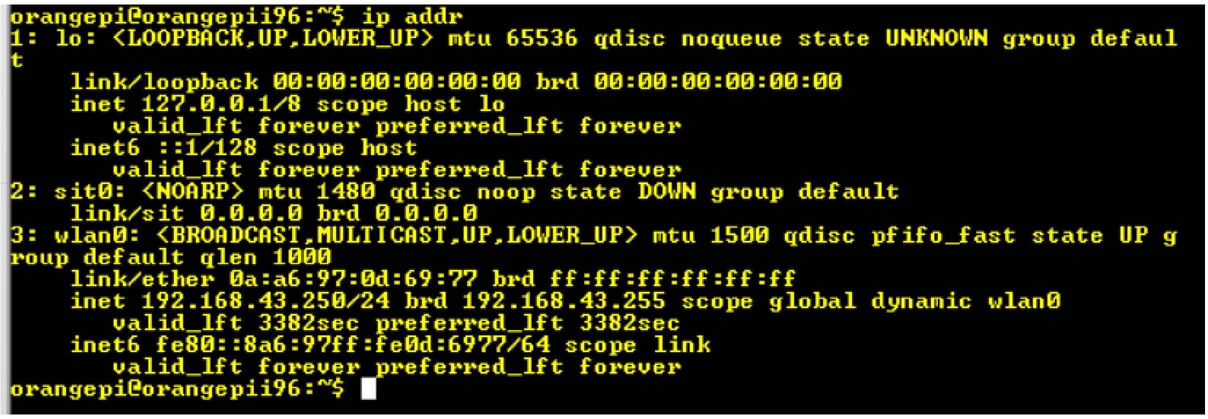
FIGURE 7 - CONSOLE DIALOG TO ENTER WI-FI AP NAME

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 WiFi" and in newly opened dialog entering WiFi AP name (SSID) and password. Doing reboot (just to be sure) of the system by using the command:

$ sudo reboot

After reboot, login to our Debian server and running a command

$ ip addr



```
orangepi@orangepii96:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: sit0: <NOARP> mtu 1480 qdisc noop state DOWN group default
    link/sit 0.0.0.0 brd 0.0.0.0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
roup default qlen 1000
    link/ether 0a:a6:97:0d:69:77 brd ff:ff:ff:ff:ff:ff
    inet 192.168.43.250/24 brd 192.168.43.255 scope global dynamic wlan0
       valid_lft 3382sec preferred_lft 3382sec
    inet6 fe80::8a6:97ff:fe0d:6977/64 scope link
       valid_lft forever preferred_lft forever
orangepi@orangepii96:~$
```

FIGURE 8 - DEBIAN CONSOLE AFTER THE COMMAND 'IP ADDR'

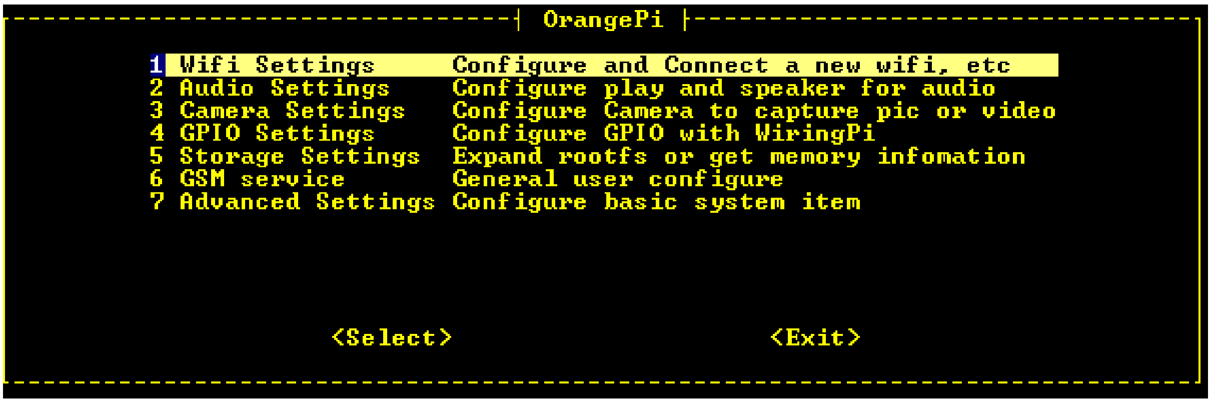
Picture above shows what IP address of our Debian server is. On our Debian server, IP address stays the same for many days independently of power on/off of the module and of the hotspot. Note: author of Debian image system what we are using, states that module's manufacturers Debian image has a bug not allowing to have stable between work sessions MAC and IP addresses.

Configuring SSH

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

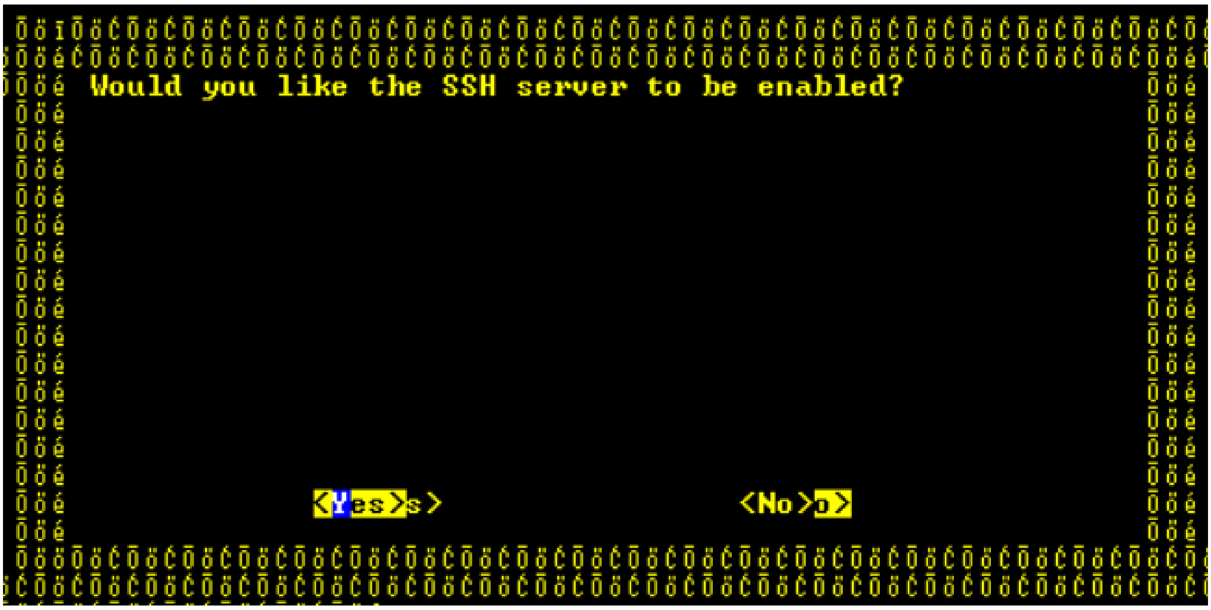
$ sudo orangepi-config

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 – we can change host name from default "orangepii96" to some custom.



```
---------------------------------┤ OrangePi ├----------------------------------

          1 Wifi Settings       Configure and Connect a new wifi, etc
          2 Audio Settings      Configure play and speaker for audio
          3 Camera Settings     Configure Camera to capture pic or video
          4 GPIO Settings       Configure GPIO with WiringPi
          5 Storage Settings    Expand rootfs or get memory infomation
          6 GSM service         General user configure
          7 Advanced Settings   Configure basic system item




                  <Select>                          <Exit>

-------------------------------------------------------------------------------
```

FIGURE 9 - UBUNTU CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG



```
   Would you like the SSH server to be enabled?




                <Yes>s>                     <No>o>
```

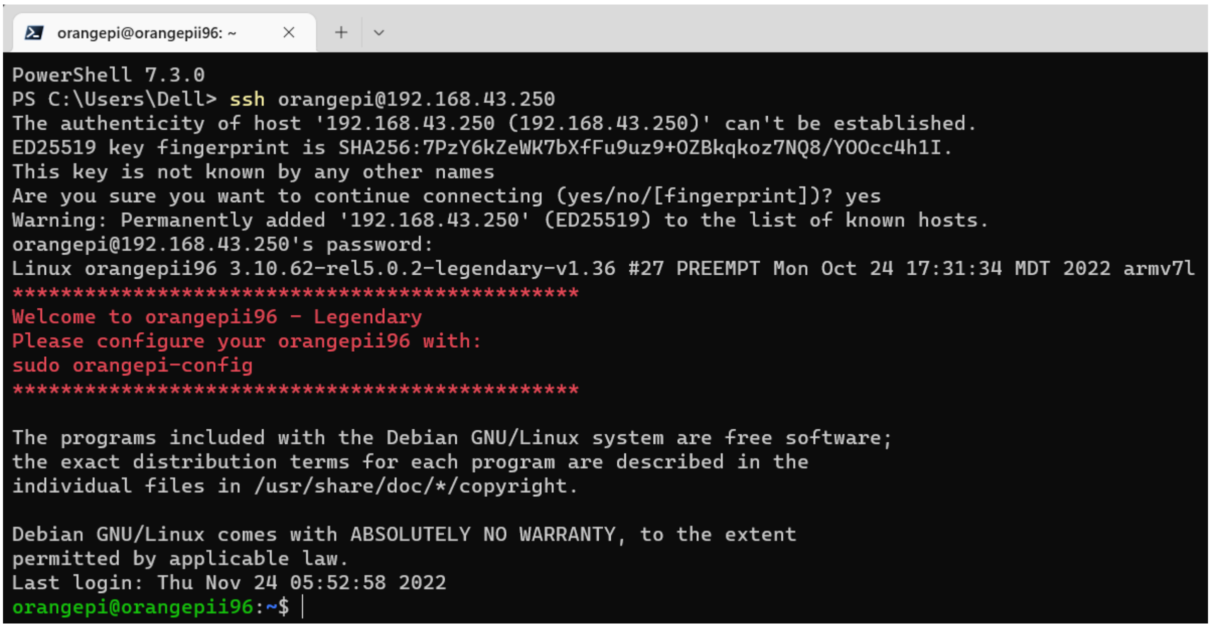
FIGURE 10 - DEBIAN CONSOLE DIALOG ENABLE SSH (UNDER ADVANCED SETTINGS)

Doing cold reboot of the system by using the command:

$ sudo halt

After disconnecting and reconnecting power, login to our Debian server 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 our server. Note2: our Windows computer's WiFi must be connected to the same WiFi hotspot as our Debian server.

Picture below shows communication session from Windows PowerShell.



```
PowerShell 7.3.0
PS C:\Users\Dell> ssh orangepi@192.168.43.250
The authenticity of host '192.168.43.250 (192.168.43.250)' can't be established.
ED25519 key fingerprint is SHA256:7PzY6kZeWK7bXfFu9uz9+OZBkqkoz7NQ8/YOOcc4h1I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.43.250' (ED25519) to the list of known hosts.
orangepi@192.168.43.250's 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 05:52:58 2022
orangepi@orangepii96:~$
```

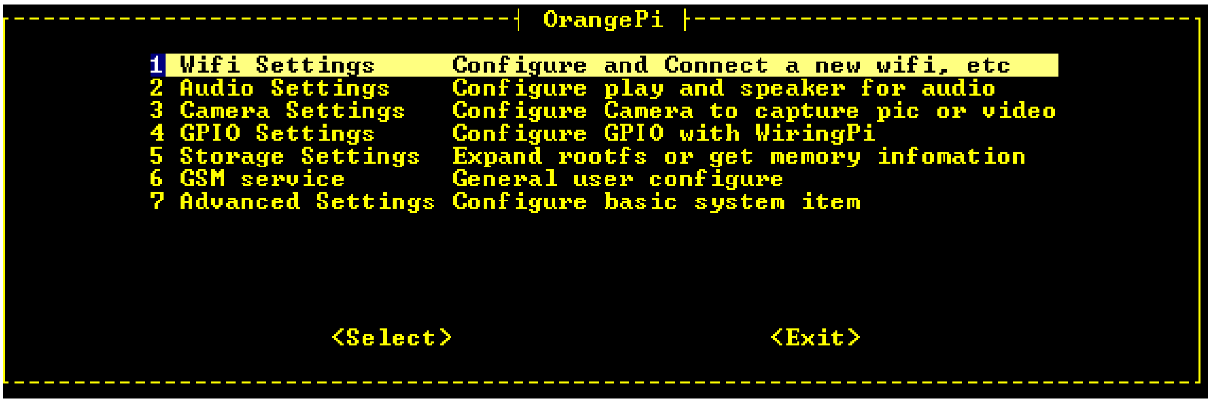
FIGURE 11 – DEBIAN SSH CONSOLE SESSION FROM WINDOWS POWERSHELL

Configuring GPIO

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

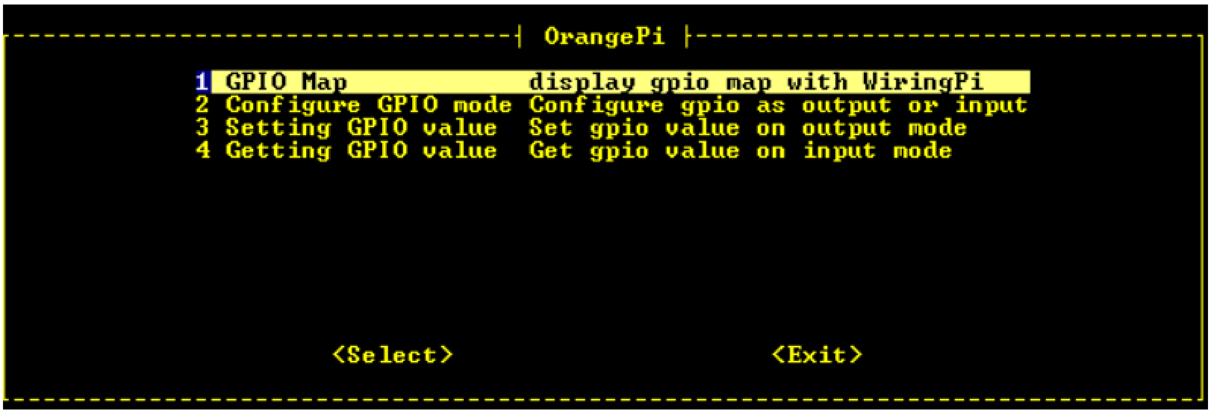
$ sudo orangepi-config

Picture below contains 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.



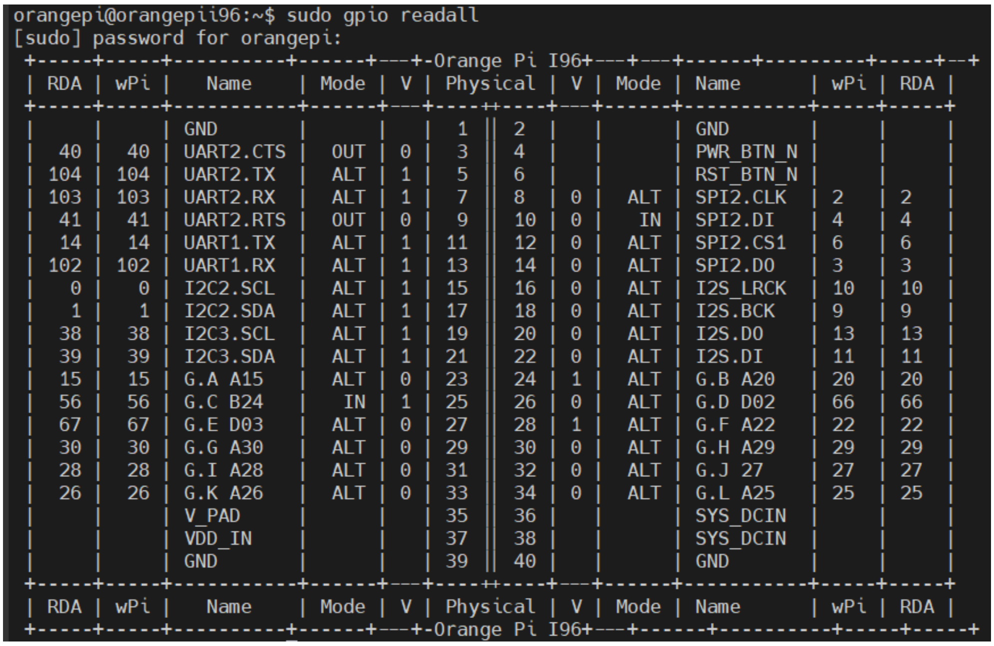
```
                    ┤ OrangePi ├

1 Wifi Settings      Configure and Connect a new wifi, etc
2 Audio Settings     Configure play and speaker for audio
3 Camera Settings    Configure Camera to capture pic or video
4 GPIO Settings      Configure GPIO with WiringPi
5 Storage Settings   Expand rootfs or get memory infomation
6 GSM service        General user configure
7 Advanced Settings  Configure basic system item


            <Select>                    <Exit>
```

FIGURE 12 - CONSOLE DIALOG AFTER SUDO ORANGEPI-CONFIG



```
----------------------------┤ OrangePi ├----------------------------

        1 GPIO Map             display gpio map with WiringPi
        2 Configure GPIO mode  Configure gpio as output or input
        3 Setting GPIO value   Set gpio value on output mode
        4 Getting GPIO value   Get gpio value on input mode




              <Select>                      <Exit>
--------------------------------------------------------------------
```

Figure 13 - 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 14 - 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 we can found working on our Debian system (**pin "56" by wiringPi naming**) – because almost all pins by default are set to do ALT function.

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

The 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

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

$ sudo apt-get update # Install the newest versions of all packages currently installed#

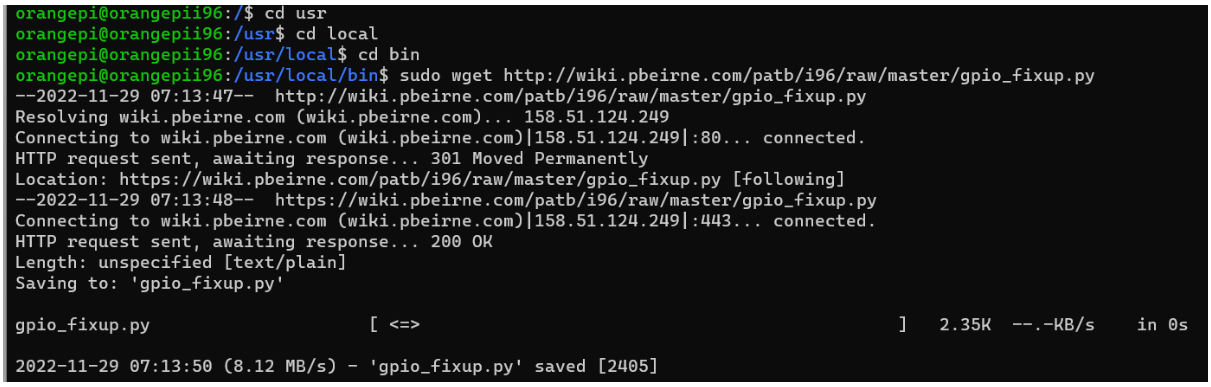
$ sudo apt-get upgrade

$ sudo apt install wget

Starting to get gpio_fixup.py:

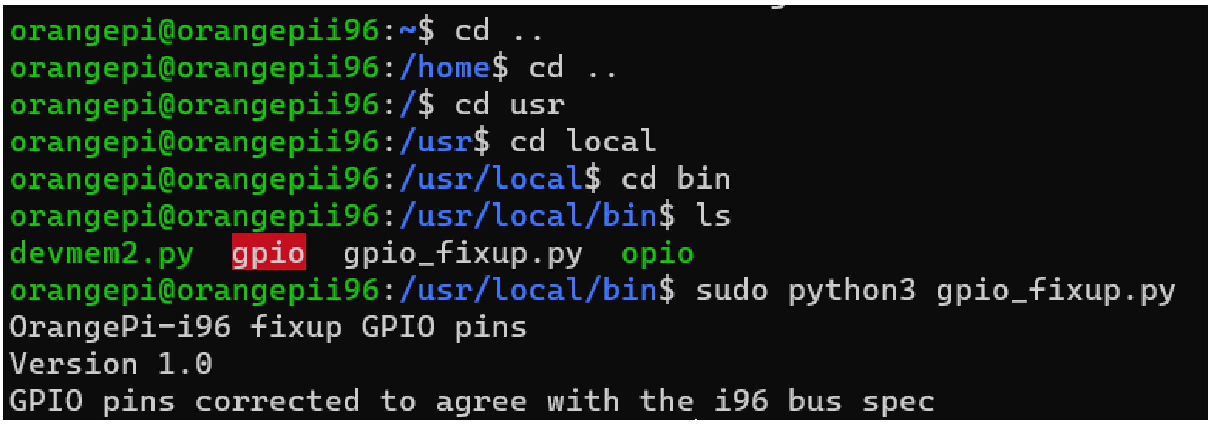
cd /usr/local/bin

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 15 - CONSOLE DIALOG GETTING GPIO_FIXUP.PY



```
orangepi@orangepii96:~$ cd ..
orangepi@orangepii96:/home$ cd ..
orangepi@orangepii96:/$ cd usr
orangepi@orangepii96:/usr$ cd local
orangepi@orangepii96:/usr/local$ cd bin
orangepi@orangepii96:/usr/local/bin$ ls
devmem2.py  gpio  gpio_fixup.py  opio
orangepi@orangepii96:/usr/local/bin$ sudo python3 gpio_fixup.py
OrangePi-i96 fixup GPIO pins
Version 1.0
GPIO pins corrected to agree with the i96 bus spec
```

FIGURE 16 - CONSOLE DIALOG EXECUTING SUDO PYTHON3 GPIO_FIXUP.PY

Next – we must make 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:

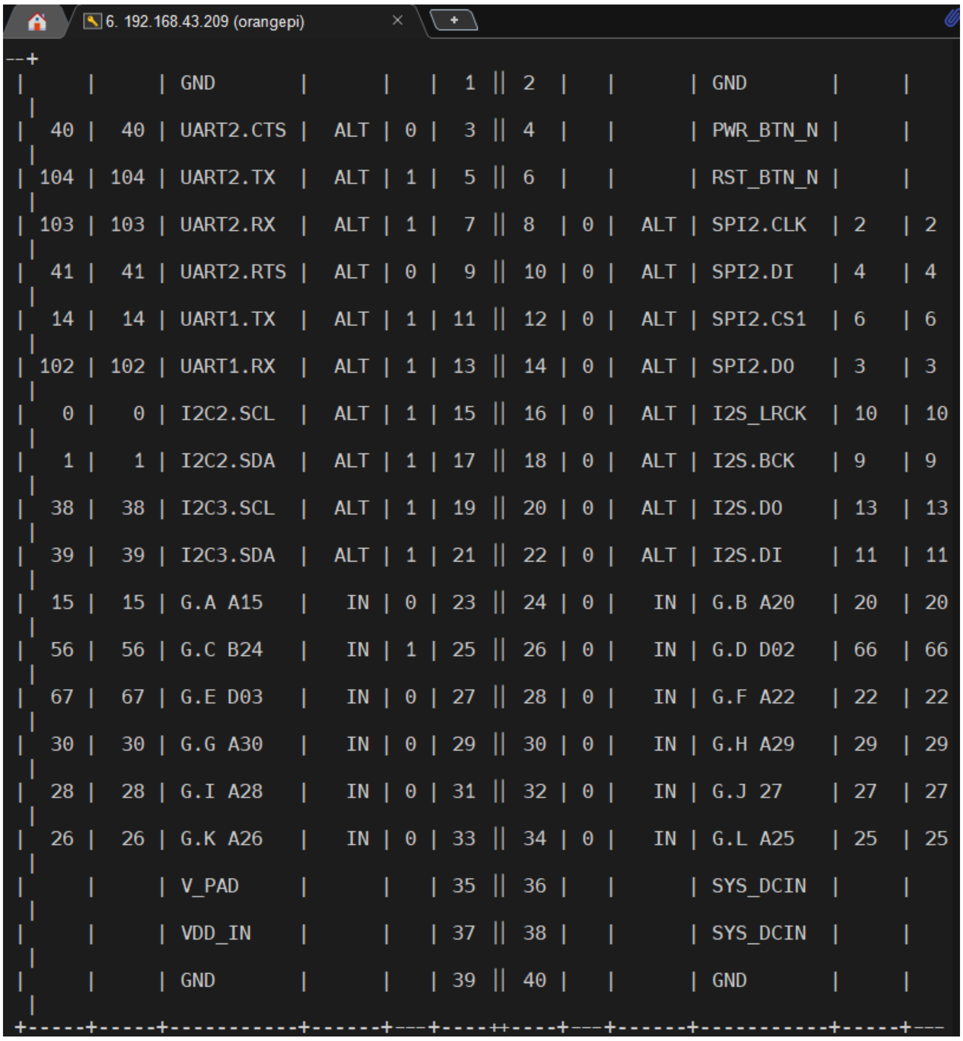
sudo sed -i "/^exit 0$/i\/usr/local/bin/gpio_fixup.py" /etc/rc.local

Modifying permissions:

$ cd /usr/local/bin

$ sudo chmod +x gpio_fixup.py

Now GPIO map is correct (run *sudo gpio readll*):



```
--+
|     |     | GND        |      |   |  1 || 2  |   |      | GND        |     |
 |
|  40 |  40 | UART2.CTS  |  ALT | 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  |  ALT | 0 |  9 || 10 | 0 |  ALT | 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    |   IN | 0 | 23 || 24 | 0 |   IN | G.B A20    | 20  | 20
 |
|  56 |  56 | G.C B24    |   IN | 1 | 25 || 26 | 0 |   IN | G.D D02    | 66  | 66
 |
|  67 |  67 | G.E D03    |   IN | 0 | 27 || 28 | 0 |   IN | G.F A22    | 22  | 22
 |
|  30 |  30 | G.G A30    |   IN | 0 | 29 || 30 | 0 |   IN | G.H A29    | 29  | 29
 |
|  28 |  28 | G.I A28    |   IN | 0 | 31 || 32 | 0 |   IN | G.J 27     | 27  | 27
 |
|  26 |  26 | G.K A26    |   IN | 0 | 33 || 34 | 0 |   IN | G.L A25    | 25  | 25
 |
|     |     | V_PAD      |      |   | 35 || 36 |   |      | SYS_DCIN   |     |
 |
|     |     | VDD_IN     |      |   | 37 || 38 |   |      | SYS_DCIN   |     |
 |
|     |     | GND        |      |   | 39 || 40 |   |      | GND        |     |
 |
+-----+-----+----------+------+---+----++----+---+------+----------+-----+---
```

FIGURE 17 - GPIO MAP ON MOBAXTERM CONSOLE DIALOG AFTER CORRECTING GPIO

Making to manage WiFi without Serial Console

We want to use our server without having Serial Console always connected. Our server must work in following automatic manner after Power ON:

1. LED indicator “WiFi status” – switch to red.
2. Check own IP address.
3. If own IP address is kind of “192.168.xxx.xxx”, send value of this address to cloud server (address of cloud server – hard coded in the script managing these procedures). Read response from cloud server. In case of success - LED indicator “WiFi status” – switch to green. End this automatic procedure.
4. If IP address is not good, look for information about saved WiFi connections. If there is no saved WiFi information, switch to mode WiFi AP. LED indicator “WiFi status” – switch to blinking blue.

We will do these automatic actions with help of some Python code. Selecting to use method of calling command line tools from Python script. Now we will install required command line tools:

Installing Node.js

Latest Node.js version now is "18".

But we will find, that the Node.js module onoff supports Node.js versions 10, 12, 14, 15 and 16. So, we are 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 we know that our processor's 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, we are 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, our 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 type of archive is "xz":

```
$ sudo apt install xz-utils
```

```
$ sudo apt-get install wget #(if not installed on some previous step)
```

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 local time zone

We must correct time zone settings:

```
$ date
```

```
Thu Nov 24 06:36:52 UTC 2022
```

```
$ sudo timedatectl list-timezones # // get list of time zones
```

Find our zone from the list.

Change time zone:

```
$ sudo timedatectl set-timezone Europe/Vilnius
```

Now our server time is:

```
Thu Nov 24 08:42:56 EET 2022
```

Installing Node-RED

It is required to have Node.js - we 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 we did this:

```
$ sudo npm install -g npm@9.2.0
```

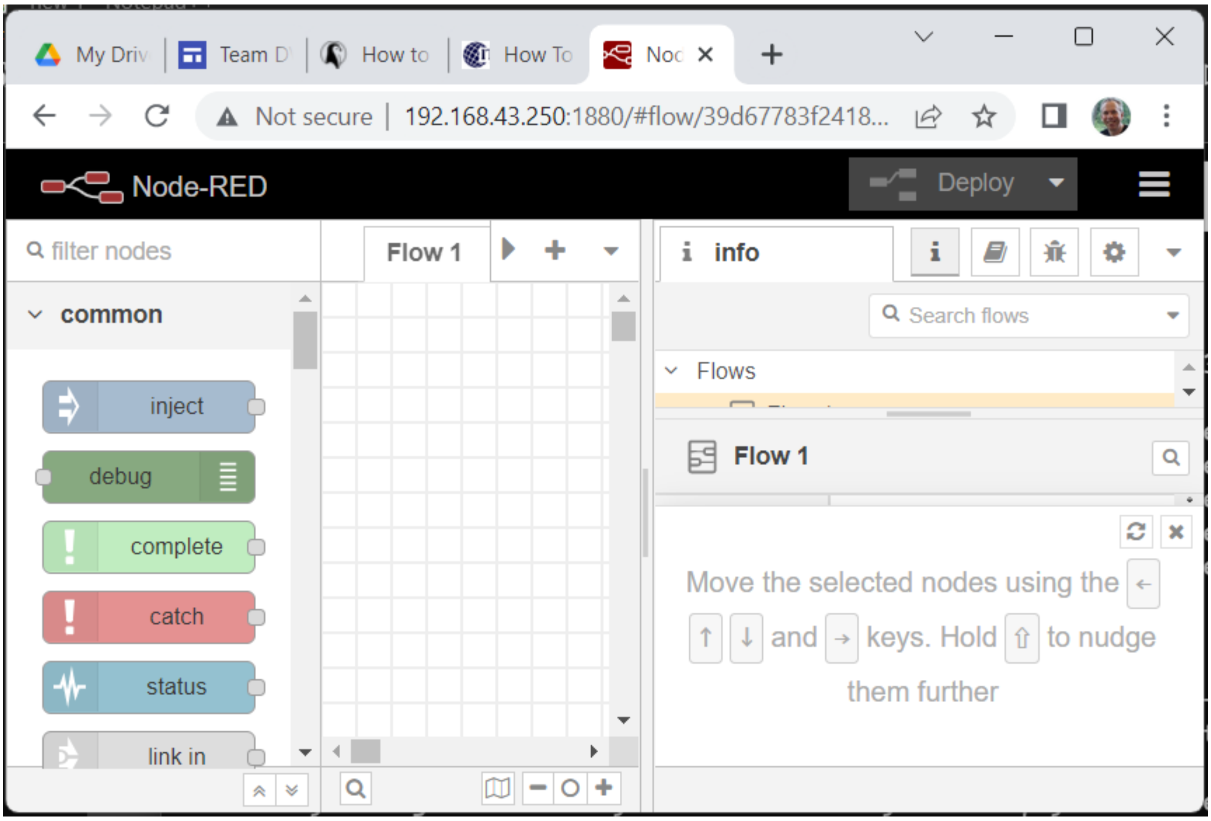
Now check how our node-red works:

```
$ 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
```



Move the selected nodes using the ← ↑ ↓ and → keys. Hold ⇧ to nudge them further

FIGURE 18 - OUR SERVER'S NODE-RED WEB PAGE ON BROWSER RUNNING ON OUR WINDOWS

Make Node-RED to work as a service

At this point the required command *node-red-start* does not exist in our system. The script below must install all required dependencies and must make all required files, but doesn't work on our system (with error "/dev/fd/63: No such file or directory"):

https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered

Now we will investigate script from above link, find required commands parts and will execute all these commands manually (use Windows PowerShell and Windows clipboard to copy-paste):

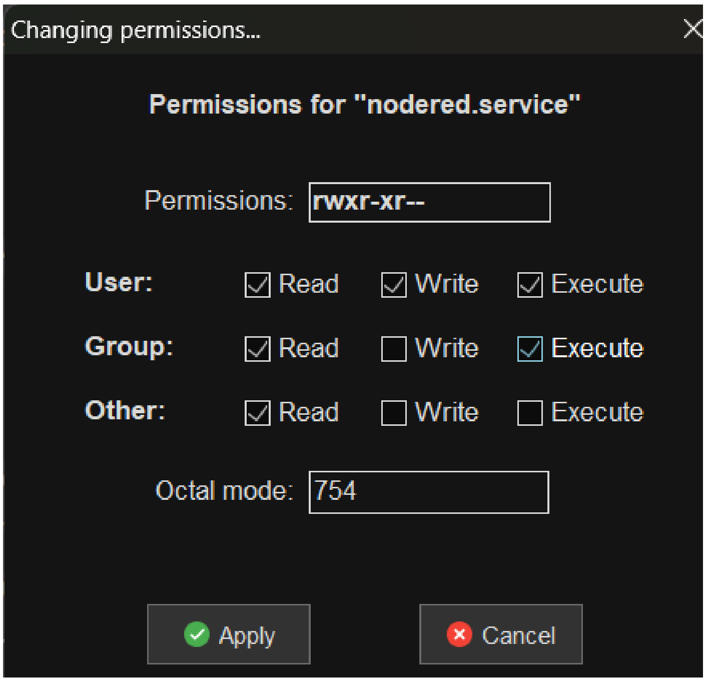
```
$ sudo curl -sL -m 60 -o /usr/bin/node-red-start https://raw.githubusercontent.com/node-red /linux-installers/master/resources/node-red-start

$ sudo curl -sL -m 60 -o /usr/bin/node-red-stop https://raw.githubusercontent.com/node-red/linux-installers/master/resources/node-red-stop

$ sudo curl -sL -m 60 -o /usr/bin/node-red-restart https://raw.githubusercontent.com/node-red/linux-installers/master/resources/node-red-restart

$ sudo curl -sL -m 60 -o /usr/bin/node-red-reload https://raw.githubusercontent.com/node-red/linux-installers/master/resources/node-red-reload

$ sudo curl -sL -m 60 -o /usr/bin/node-red-log https://raw.githubusercontent.com/node-red/linux-installers/master/resources/node-red-log

$ sudo curl -sL -m 60 -o /etc/logrotate.d/nodered https://raw.githubusercontent.com/node-red/linux-installers/master/resources/nodered.rotate

$ sudo chmod +x /usr/bin/node-red-start

$ sudo chmod +x /usr/bin/node-red-stop

$ sudo chmod +x /usr/bin/node-red-restart

$ sudo chmod +x /usr/bin/node-red-reload

$ sudo chmod +x /usr/bin/node-red-log

$ sudo curl -sL -m 60 -o /usr/share/icons/hicolor/scalable/apps/node-red-icon.svg https://raw.githubusercontent.com/node-red/linux-installers/master/resources/node-red-icon.svg
```

Now we need the file nodered.service located at /etc/systemd/system/ and with correct settings in it. We are downloading this file from the link https://raw.githubusercontent.com/node-red/linux-installers/master/resources/nodered.service

On our Windows computer, name downloaded file to *nodered.service,* save it locally and edit by making following changes:

```
User=orangepi

Group=orangepi

WorkingDirectory=/home/orangepi

Environment="NODE_OPTIONS=--max_old_space_size=256"
```

After, we will upload edited file by using MobaXterm from Windows file system to our server’s directory /home/orangepi/. We are uploading this file to home directory, not to system directory, because we don’t have permissions from MobaXterm browser to paste this file and change permissions in system directory. When our file nodered.service appears in our home directory, we change permissions of the file nodered.service to allow execute like in picture bellow:



Changing permissions...

Permissions for "nodered.service"

Permissions: rwxr-xr--

User: ☑ Read ☑ Write ☑ Execute

Group: ☑ Read □ Write ☑ Execute

Other: ☑ Read □ Write □ Execute

Octal mode: 754

FIGURE 19 - ALLOWING EXECUTE NODERED.SERVICE FILE

Now we must copy this file to system directory. Locate our console to home directory and execute the commands:

```
$ sudo cp nodered.service /etc/systemd/system
```

And try to start nodered.service:

```
$ node-red-start  // # just to check
```

Make to work as a service:

```
$ sudo systemctl enable nodered.service
```

```
$ sudo reboot
```

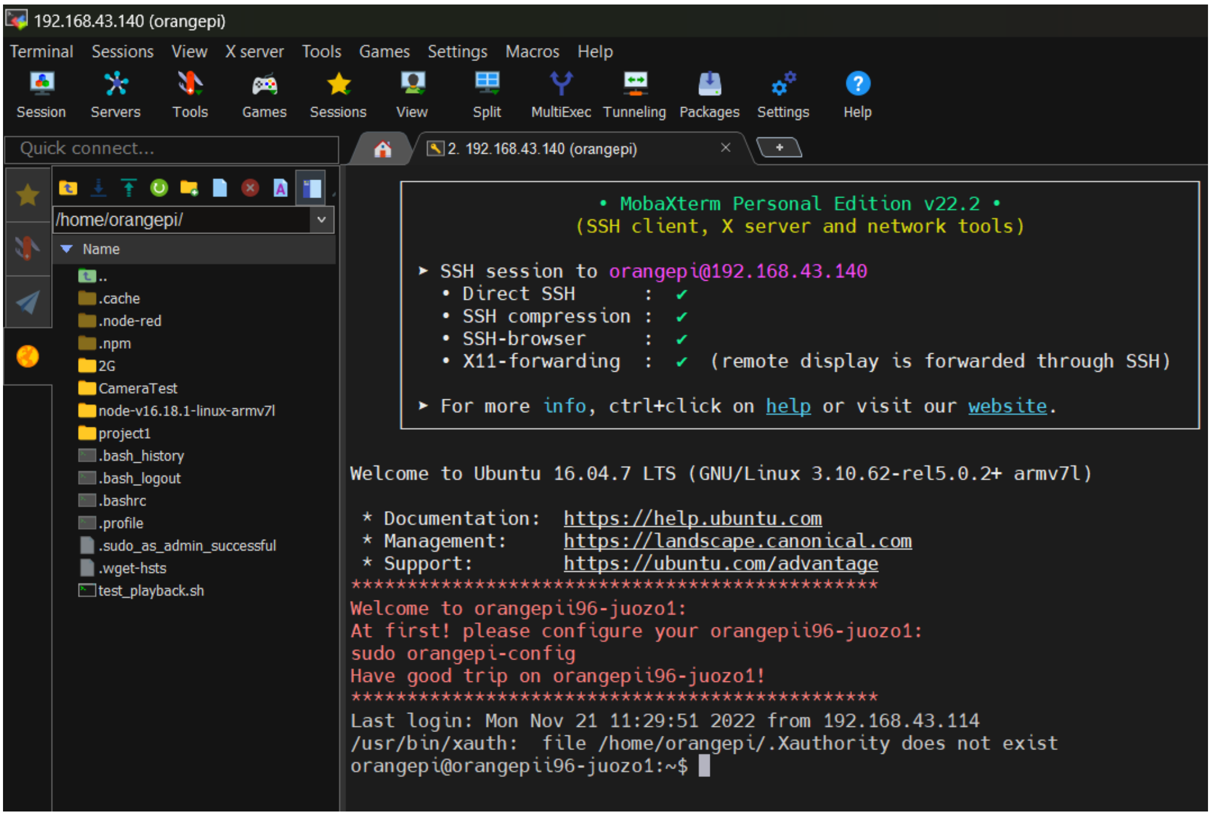
Checking on Windows computer browser address http://192.168.43.94:1880. We see that node-red service works!

Open and edit files with MobaXterm

It will be very helpful to open and edit on our Windows computer text and code files located inside of our Debian server. Checking how works the program Sublime Text 4 (with all required settings on our Debian) and finding this program working and useful. But Sublime Text 4 is not free to use to anyone. After we are finding other program, which is free for home using and contains impressive additional function to remote browse files and also many other good functions.

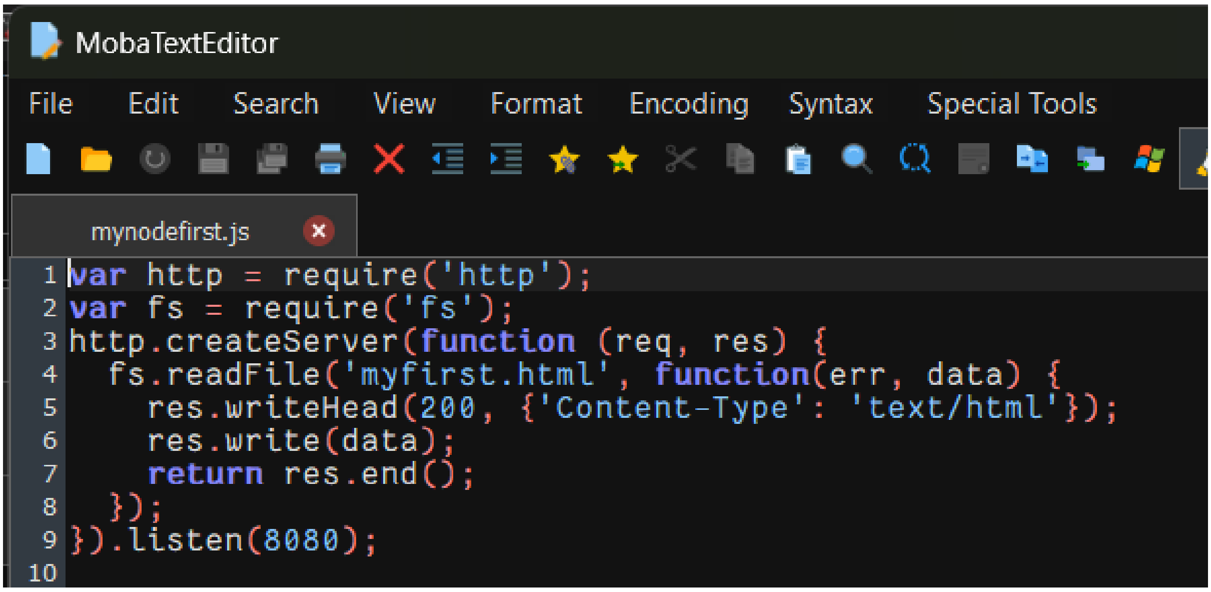
Download page is this: https://mobaxterm.mobatek.net/download-home-edition.html

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 Debian server installation (in opposite to Sublime Text 4).


192.168.43.140 (orangepi)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 192.168.43.140 (orangepi)
/home/orangepi/
Name
..
.cache
.node-red
.npm
2G
CameraTest
node-v16.18.1-linux-armv7l
project1
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.wget-hsts
test_playback.sh

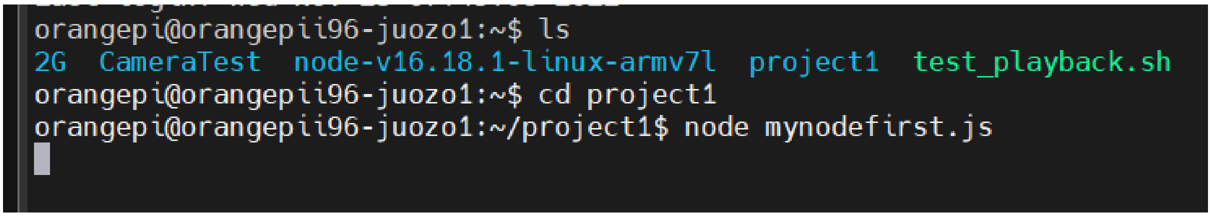
```
          • MobaXterm Personal Edition v22.2 •
        (SSH client, X server and network tools)

 ➤ SSH session to orangepi@192.168.43.140
   • Direct SSH      :  ✔
   • SSH compression :  ✔
   • SSH-browser     :  ✔
   • X11-forwarding  :  ✔  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 3.10.62-rel5.0.2+ armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
***********************************************
Welcome to orangepii96-juozo1:
At first! please configure your orangepii96-juozo1:
sudo orangepi-config
Have good trip on orangepii96-juozo1!
***********************************************
Last login: Mon Nov 21 11:29:51 2022 from 192.168.43.114
/usr/bin/xauth:  file /home/orangepi/.Xauthority does not exist
orangepi@orangepii96-juozo1:~$
```

FIGURE 20 - MOBAXTERM PROGRAM WINDOW AFTER CONNECTION TO OUR DEBIAN SERVER



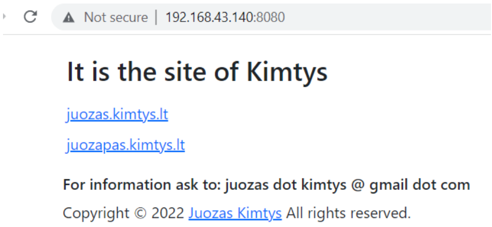
```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('myfirst.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    return res.end();
  });
}).listen(8080);
```

FIGURE 21 - EDITING OUR NODE.JS PROGRAM ON MOBATEXTEDITOR



```
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 22 - RUNNING OUR NODE.JS PROGRAM



It is the site of Kimtys

juozas.kimtys.lt

juozapas.kimtys.lt

For information ask to: juozas dot kimtys @ gmail dot com

Copyright © 2022 Juozas Kimtys All rights reserved.

FIGURE 23 - WEB PAGE EXAMPLE GENERATED BY NODE.JS ON OUR ORANGEPI I96 SERVER

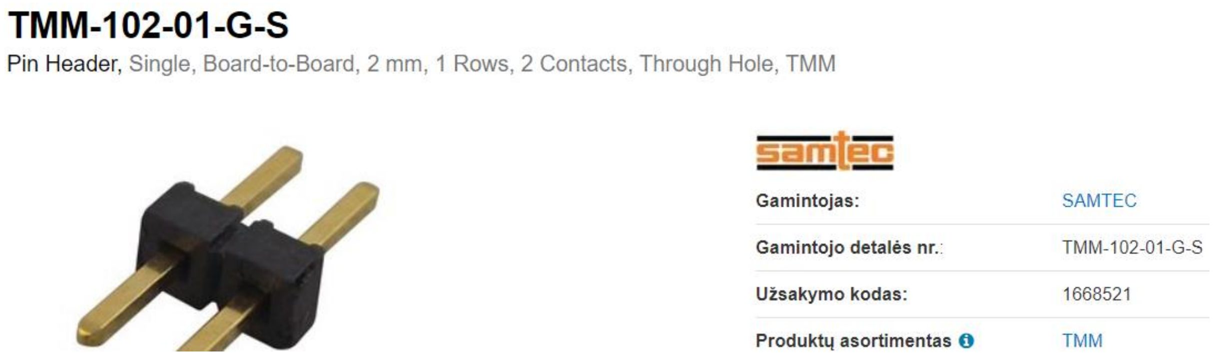
Blink LED (Node.js)

In order to toggle on-off GPIO pins, we 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



TMM-102-01-G-S

Pin Header, Single, Board-to-Board, 2 mm, 1 Rows, 2 Contacts, Through Hole, TMM

Gamintojas:	SAMTEC
Gamintojo detalės nr.:	TMM-102-01-G-S
Užsakymo kodas:	1668521
Produktų asortimentas ℹ	TMM

FIGURE 24 - FINDING CONNECTORS HAVING PITCH 2.0MM (SCREENSHOT FROM FARNELL)

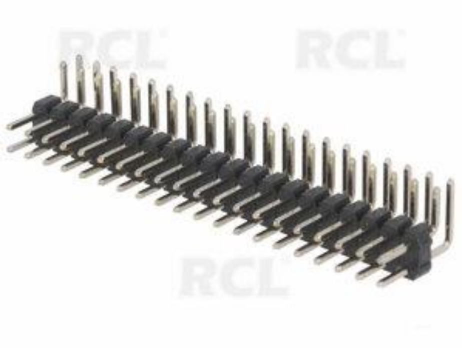


FIGURE 25 - FINDING CONNECTORS HAVING PITCH 2.0MM (CJK4202K/2 FROM RCL.LT)

Connecting oscilloscope probes to 40-pins GPIO connector. Initially we are 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 we configured previously), all pins will be accessible.

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

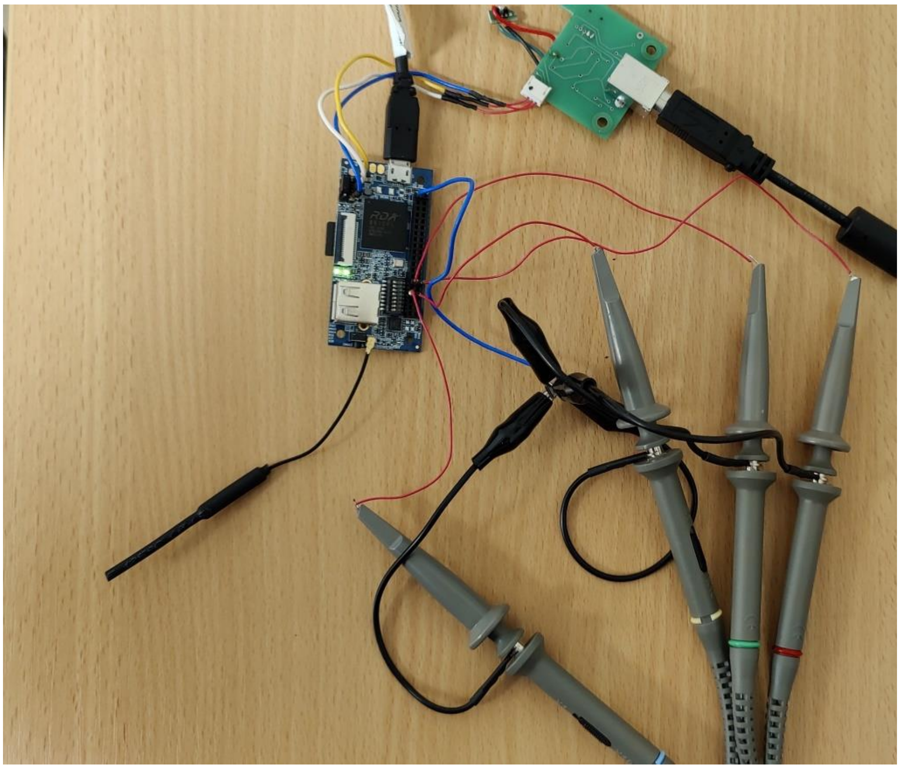


FIGURE 26 - CONNECTING OSCILLOSCOPE PROBES TO SOME GPIO PINS BY USING 2-PINS CONNECTORS

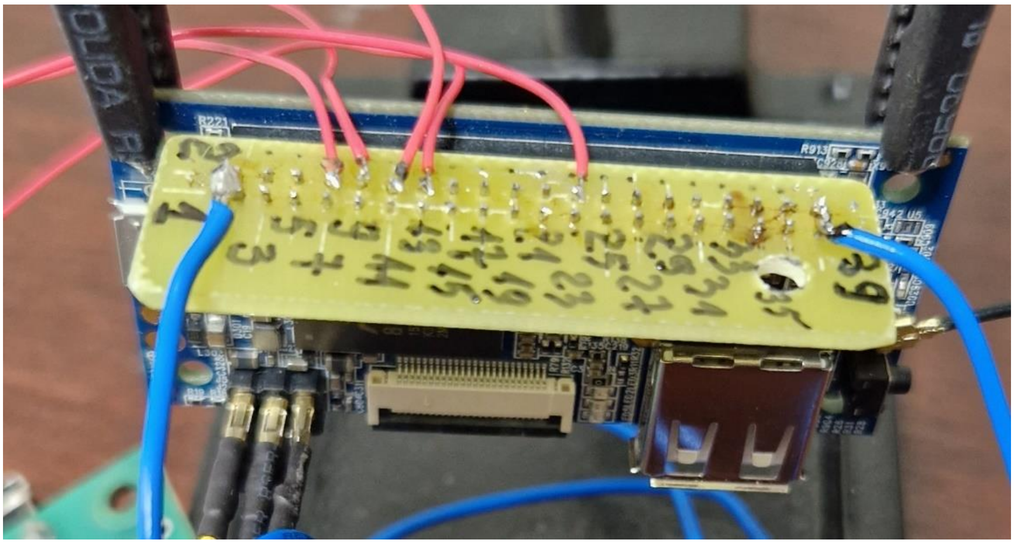
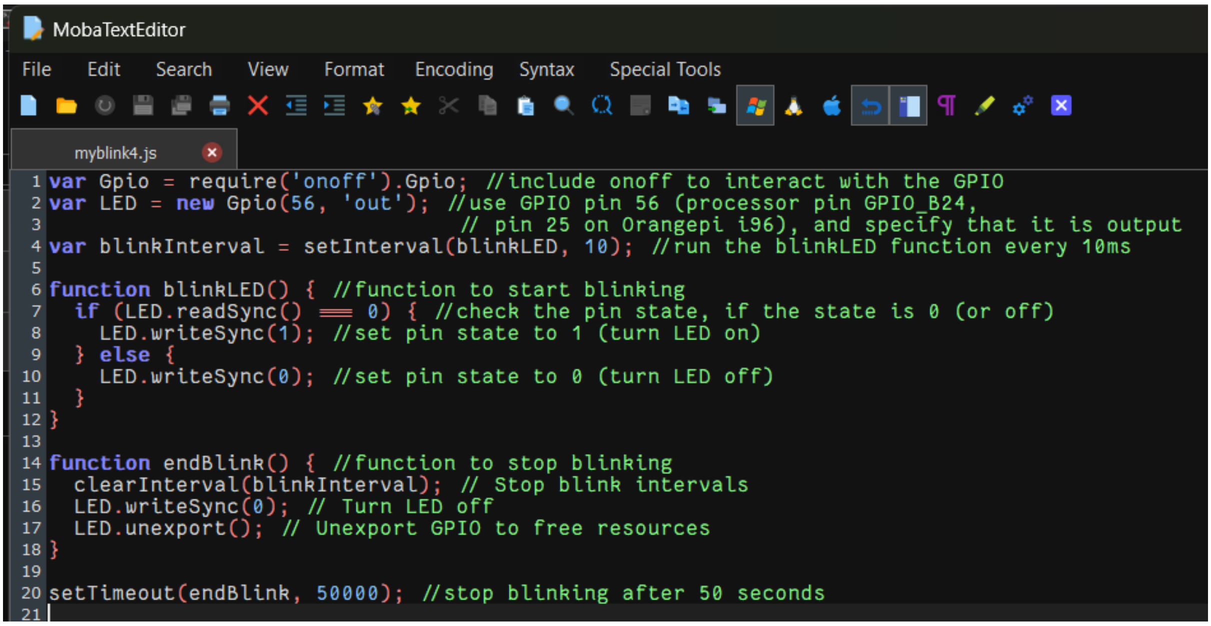


FIGURE 27 - PROTOTYPE BOARD TO CONNECT OSCILLOSCOPE PROBES BY USING 40-PINS CONNECTOR



```
var Gpio = require('onoff').Gpio; //include onoff to interact with the GPIO
var LED = new Gpio(56, 'out'); //use GPIO pin 56 (processor pin GPIO_B24,
                               // pin 25 on Orangepi i96), and specify that it is output
var blinkInterval = setInterval(blinkLED, 10); //run the blinkLED function every 10ms

function blinkLED() { //function to start blinking
  if (LED.readSync() === 0) { //check the pin state, if the state is 0 (or off)
    LED.writeSync(1); //set pin state to 1 (turn LED on)
  } else {
    LED.writeSync(0); //set pin state to 0 (turn LED off)
  }
}

function endBlink() { //function to stop blinking
  clearInterval(blinkInterval); // Stop blink intervals
  LED.writeSync(0); // Turn LED off
  LED.unexport(); // Unexport GPIO to free resources
}

setTimeout(endBlink, 50000); //stop blinking after 50 seconds

```

FIGURE 28 - MAKING NODE.JS CODE ON MOBAXTERM MOBATEXTEDITOR

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

$ sudo node myblink4.js

We need to do "sudo" because there are required permissions to write pins.

It is possible to stop execution earlier than code ends - by using CTRL+Z

Using I2C ADC (Node.js)

To make measurement with I2C ADC, we will use the module " i2c-bus" : https://www.npmjs.com/package/i2c-bus

$ npm install i2c-bus

Making schematic as shown in datasheet of ADC chip (https://www.microchip.com/en-us/product/MCP3425) :

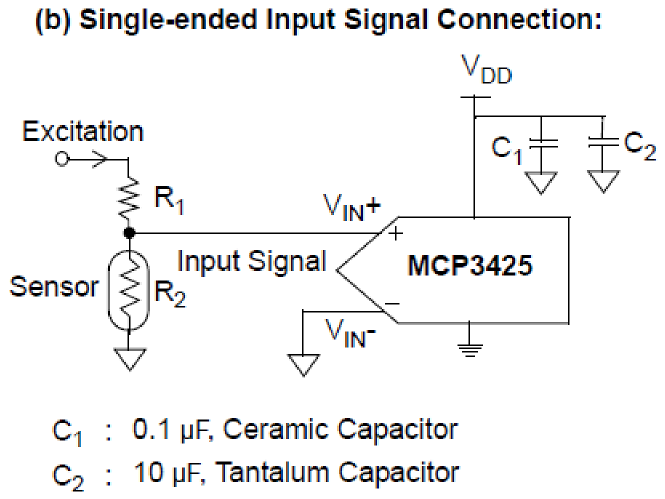

(b) Single-ended Input Signal Connection:
V_{DD}
Excitation
C_1
C_2
R_1
$V_{IN}+$
+
Input Signal
MCP3425
Sensor
R_2
–
$V_{IN}-$
C_1 : 0.1 µF, Ceramic Capacitor
C_2 : 10 µF, Tantalum Capacitor

FIGURE 29 - ANALOG CONNECTIONS OF ADC CHIP. EXCITATION FROM 3V3, TERMISTOR 10K NTC.

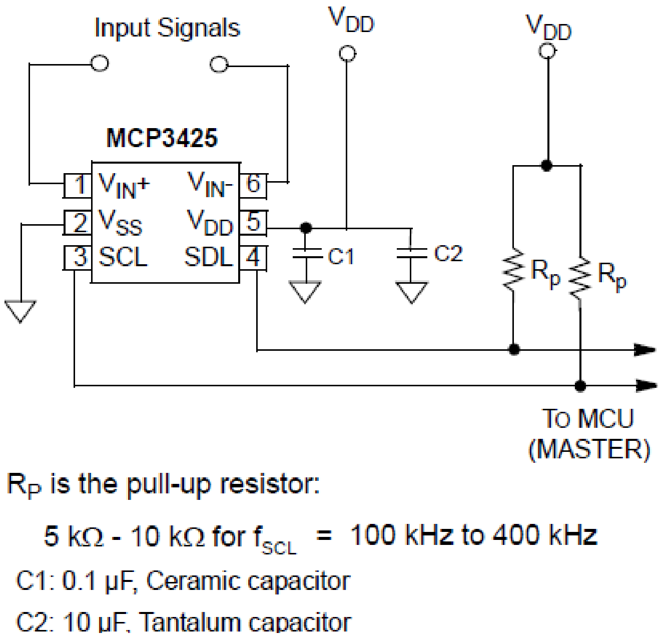

Input Signals
V_{DD}
V_{DD}
MCP3425
1 $V_{IN}+$
$V_{IN}-$ 6
2 V_{SS}
V_{DD} 5
3 SCL
SDL 4
C1
C2
R_P
R_P
To MCU
(MASTER)
R_P is the pull-up resistor:
5 kΩ - 10 kΩ for f_{SCL} = 100 kHz to 400 kHz
C1: 0.1 µF, Ceramic capacitor
C2: 10 µF, Tantalum capacitor

FIGURE 30 - DIGITAL CONNECTIONS TO ADC CHIP. RESISTORS 5K6

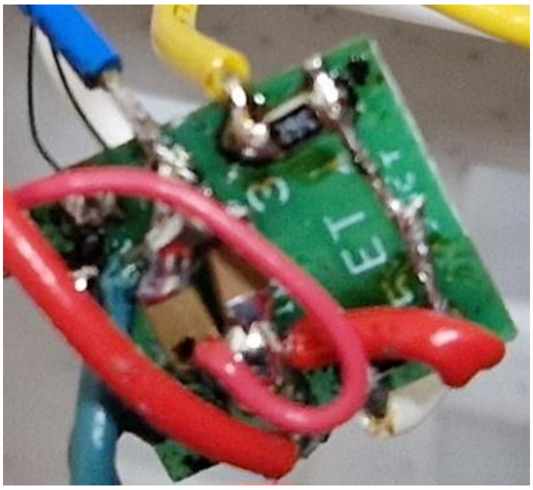
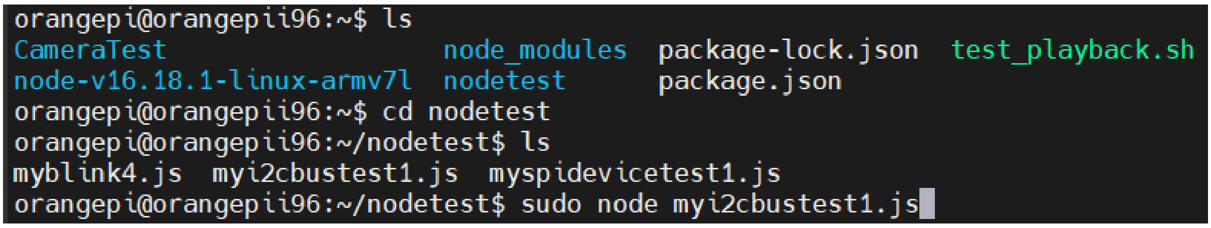


Figure 31 - Prototyping of ADC on the prototype board “SOT23-6 to DIP8”

Using Visual Studio Code editor and MobaXterm Editor. Creating the code:

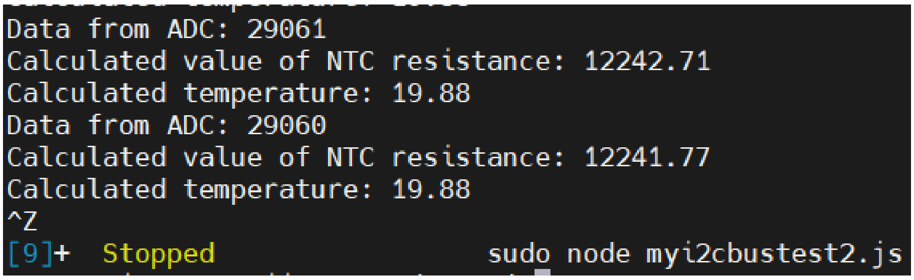
```
/*
myi2cbustest2.js
Juozas Kimtys 2022.12.21
using example from:
https://www.npmjs.com/package/i2c-bus
selecting example version "Example 3 - Asynchronous Callbacks"
our ADC chip type is MCP3425A0T
Writing settings to chip before starting to read data.
If to not write settings, default is: 12 bits (including sign) - max positive value in single ended mode is 2047,
 - max negative value in single ended mode is -2048,
max sample rate= 240 SPS, gain=1, continuous conversion
*/

const i2c = require('i2c-bus');

const MCP3425_ADDR = 0x68;
const MCP3425_MODE_BIT_CONV_MODE = 0b00010000; // 1 = Continuous Conversion Mode.
const MCP3425_MODE_BITS_SAMPLE_RATE = 0b00001000; // 10 = 15 SPS (16 bits)
const MCP3425_MODE_BITS_GAIN = 0b00000000; // 00 = 1 V/V,
const MCP3425_MODE_BITS = MCP3425_MODE_BIT_CONV_MODE | MCP3425_MODE_BITS_SAMPLE_RATE | MCP3425_MODE_BITS_GAIN;
var blnFlagModeSettingsOfAdcChipIsDone = false;

console.log('Starting i2c-bus test');

myFunc_make_ADC_chip_mode_settings();
var measurementInterval1 = setInterval(myFunc_make_measurement1, 200);

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

```
function myFunc_make_measurement1() {

const i2c1 = i2c.open(1, err => { // connected wires to pins 15 and 17 of 40-
pins GPIO connector
  if (err) throw err;
  const rawData = Buffer.alloc(3);

  i2c1.i2cRead(MCP3425_ADDR, 3, rawData, (err) => {
    if (err) throw err;
    var intMyResultFromADC = rawData[0] * 256;
    intMyResultFromADC += rawData[1];
    console.log('Data from ADC: %i', intMyResultFromADC);
   // console.log('Status byte of ADC: %i', rawData[2]);
    calculated_temperature =
myFunc_calculate_Celcius_from_ADC_data(intMyResultFromADC);
    calculated_temperature = calculated_temperature.toFixed(2);
    console.log('Calculated temperature: %d', calculated_temperature);

    i2c1.close(err => {
      if (err) throw err;
    }); //  i2c1.close
    }); //  i2c1.i2cRead(
}); // const i2c1 = i2c.open(1,
} // function myFunc_make_measurement1

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

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

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

function myFunc_make_ADC_chip_mode_settings() {
const i2c1 = i2c.open(1, err => { // connected wires to pins 15 and 17 of 40-
pins GPIO connector
    if (err) throw err;
    const rawData = Buffer.alloc(3);
    rawData[0] = MCP3425_MODE_BITS;
    i2c1.i2cWrite(MCP3425_ADDR, 1, rawData, (err) => {
      if (err) throw err;
      console.log('Make settings of ADC chip.');
      i2c1.close(err => {
        if (err) throw err;
        blnFlagModeSettingsOfAdcChipIsDone = true;
      }); //  i2c1.close
    }); //  i2c1.i2cWrite(
  }); // const i2c1 = i2c.open(1,
}

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

function myFunc_calculate_Celcius_from_ADC_data(intADCdata) {
var BValue = 3450;
var R1 = 10000;
var T1 = 298.15;
var R2 ;
var T2 ;

var a ;
var b ;
var c ;
var d ;
var e = 2.718281828 ;
var intMaxValueOfADC = 32767; // 16-bits, but including polarity sign
var dblReferenceVoltage = 2048; // internal reference of the MCP3425
var dblMeasurementVoltage = 3300;

var dblVoltageOnThermistor = intADCdata * (dblReferenceVoltage /
intMaxValueOfADC);
if ((dblMeasurementVoltage - dblVoltageOnThermistor) == 0)
{dblVoltageOnThermistor = dblMeasurementVoltage - 0.01;} // try to avoid
overflow in case if thermistor is not connected
var R2 = ((R1 * dblMeasurementVoltage) / (dblMeasurementVoltage -
dblVoltageOnThermistor)) - R1;

 console.log('Calculated value of NTC resistance: %d', R2.toFixed(2));

a = 1/T1;
b = Math.log10(R1/R2);
c = b / Math.log10(e);
d = c / BValue ;
T2 = 1 / (a- d);

return (T2 - 273.15);
}

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



```
orangepi@orangepii96:~$ ls
CameraTest                node_modules  package-lock.json  test_playback.sh
node-v16.18.1-linux-armv7l  nodetest      package.json
orangepi@orangepii96:~$ cd nodetest
orangepi@orangepii96:~/nodetest$ ls
myblink4.js  myi2cbustest1.js  myspidevicetest1.js
orangepi@orangepii96:~/nodetest$ sudo node myi2cbustest1.js
```

FIGURE 32 - CONSOLE WINDOW SHOWING RUN TEST OF I2C BUS



```
Data from ADC: 29061
Calculated value of NTC resistance: 12242.71
Calculated temperature: 19.88
Data from ADC: 29060
Calculated value of NTC resistance: 12241.77
Calculated temperature: 19.88
^Z
[9]+  Stopped                 sudo node myi2cbustest2.js
```

FIGURE 33 - CONSOLE WINDOW SHOWING RESULT OF CODE TO TEST I2C

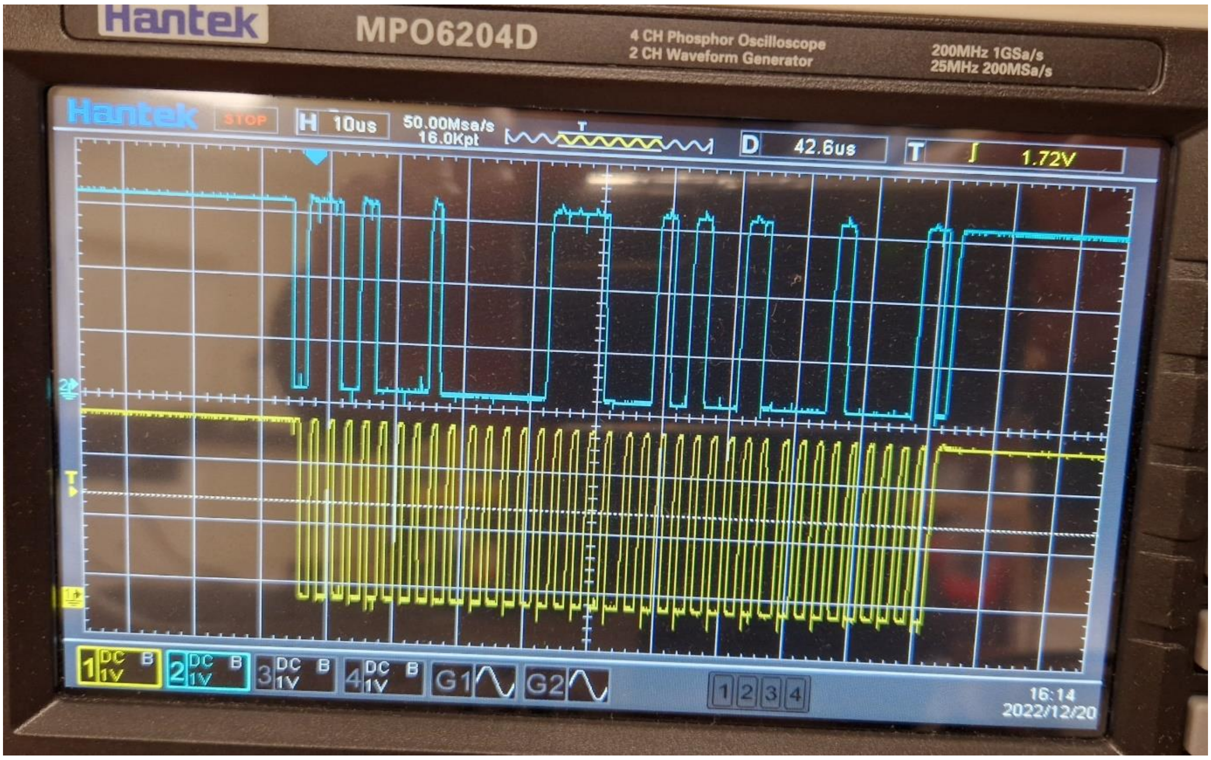


FIGURE 34 - SHAPES OF I2C BUS LINES SDA AND SCL ON OSCILLOSCOPE. WE SEE THAT CLOCK IS 400KHZ

Not success of using SPI ADC (Node.js)

To make measurement with SPI ADC, we will use the module "spi-device" : https://www.npmjs.com/package/spi-device/v/3.1.2 , SPI bus works, but we will observe on oscilloscope by looking to signal SPI-SYNC, that it is 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 is not really implemented for this type of chip (RDA8810PL). Our ADC need to change speed to some 100.000Hz or less – it won’t work at speed 20.000.000Hz.

Our code just for information:

```
/*
Test ADC of type MCP3208 on Debian 11 by using Node.js
Juozas Kimtys 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);

// We will not use special pin !SPI2_CS, because the library 'spi-device' do
not have possibility to switch to other CS pin (in order to use more than one
SPI device on the 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)
// our test board contains soldered EEPROM chip on the same SPI bus.
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) => { //(select ADC)
            if (err) throw err;
// The MCP3208 is on bus 1 and it's device 0
const mcp3208 = 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, but this
setting really does not work on the board OrangePi i96 on Debian, we will try
to set it soon separately
    }];
    if (err) throw err;
    if (blnFlagSpiOptionsNeedToBeSet)
    {
    mcp3208.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
          }]
          mcp3208.setOptions(myNewOptions, (err) => {
            if (err) throw err;
            blnFlagSpiOptionsNeedToBeSet = false;
          })
    });
    } //   if (blnFlagSpiOptionsNeedToBeSet)
    mcp3208.transfer(message, (err, message) => {
      if (err) throw err;
      const rawValue = ((message[0].receiveBuffer[1] & 0x0F) << 8) +
        message[0].receiveBuffer[2];
      console.log(intChannelCodeCurrent, ' : ', rawValue);
      intChannelCodeCurrent += 1;
      if (intChannelCodeCurrent == (intChannelCodeInitial +
intCountOfChannels))
      {intChannelCodeCurrent = intChannelCodeInitial;}
    }); // mcp3208.transfer(message
  }); // const mcp3208 = 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
```

Revision History

Version	Date	Comments
ver.2.0.1	2023.02.02	Minor spelling corrections.
ver.2.0	2022.12.20	○ Change the person of writing from "I’m doing" to "we have to do" or to neutral. Keep only minimal information about not success to use Ubuntu.
ver.1.0	2022.12.14	Initial release