www.wireless-road.com

Building Armbian image for GW-01 from sources.

Compiling image from sources.

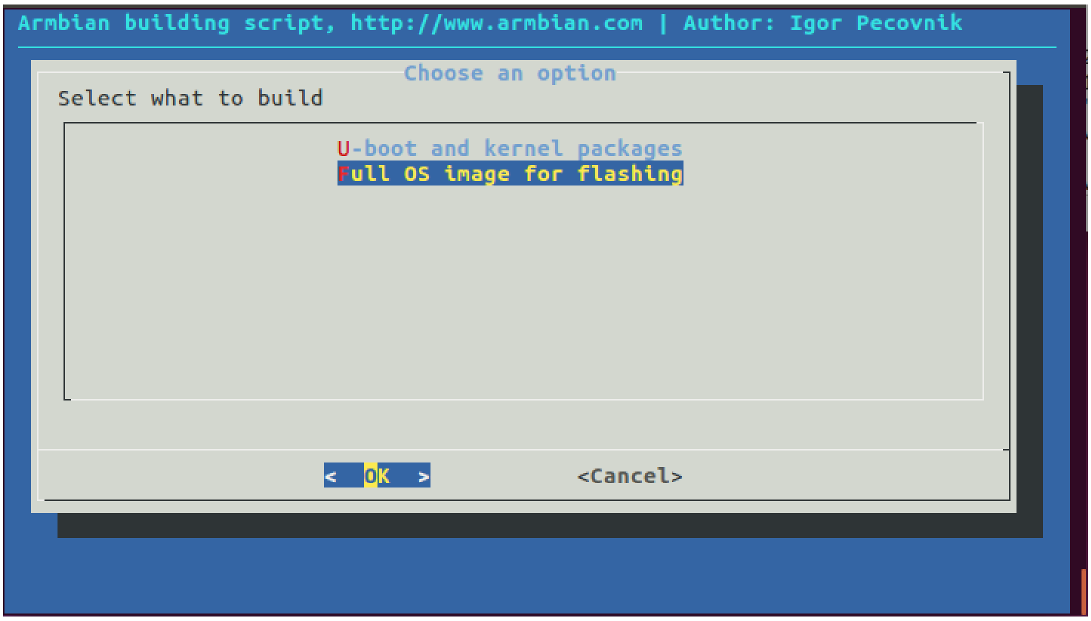
Clone Armbian OS project:

```
git clone --depth 1 https://github.com/armbian/build
cd build
```

Run following command to compile image.

```
./compile.sh
```

But before compilation process following window will appears:



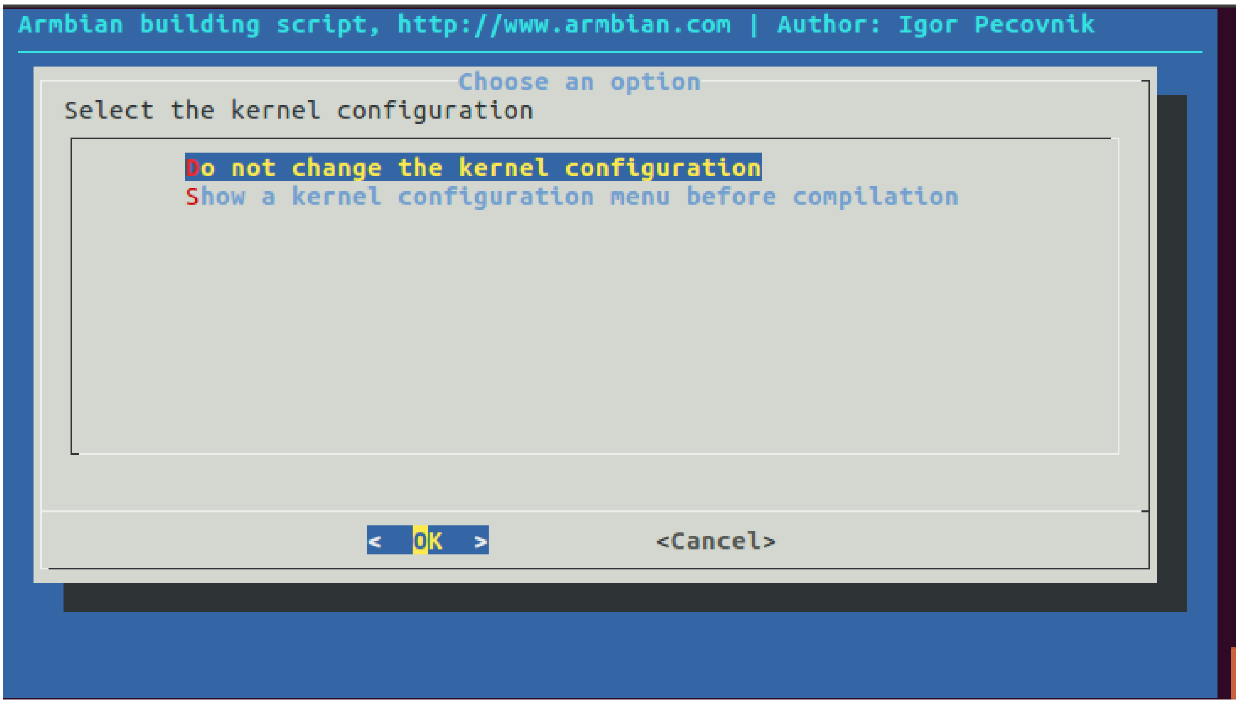
Armbian building script, http://www.armbian.com | Author: Igor Pecovnik

Choose an option

Select what to build

U-boot and kernel packages
Full OS image for flashing

Select «`Full OS image for flashing`» and press «`Ok`». In following choose «`Do not change the kernel configuration`»:



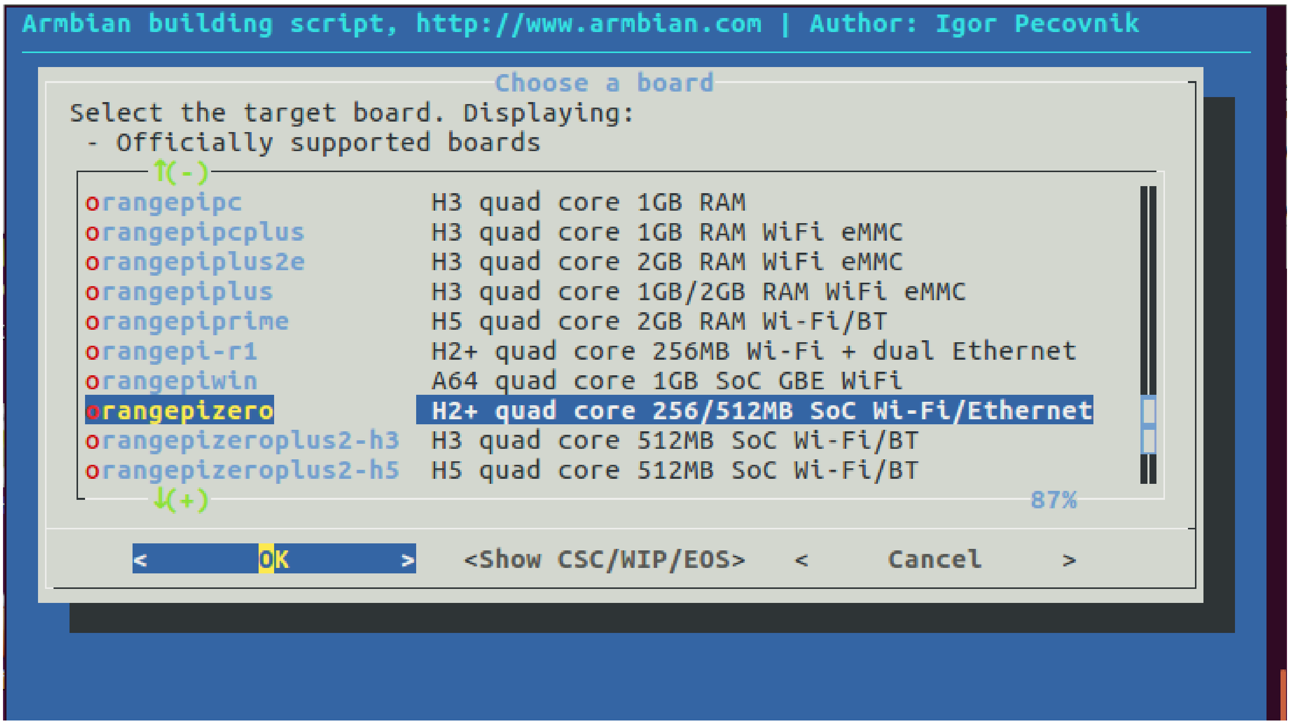
Armbian building script, http://www.armbian.com | Author: Igor Pecovnik

Choose an option

Select the kernel configuration

Do not change the kernel configuration
Show a kernel configuration menu before compilation

Then select «orangepizero» configuration:

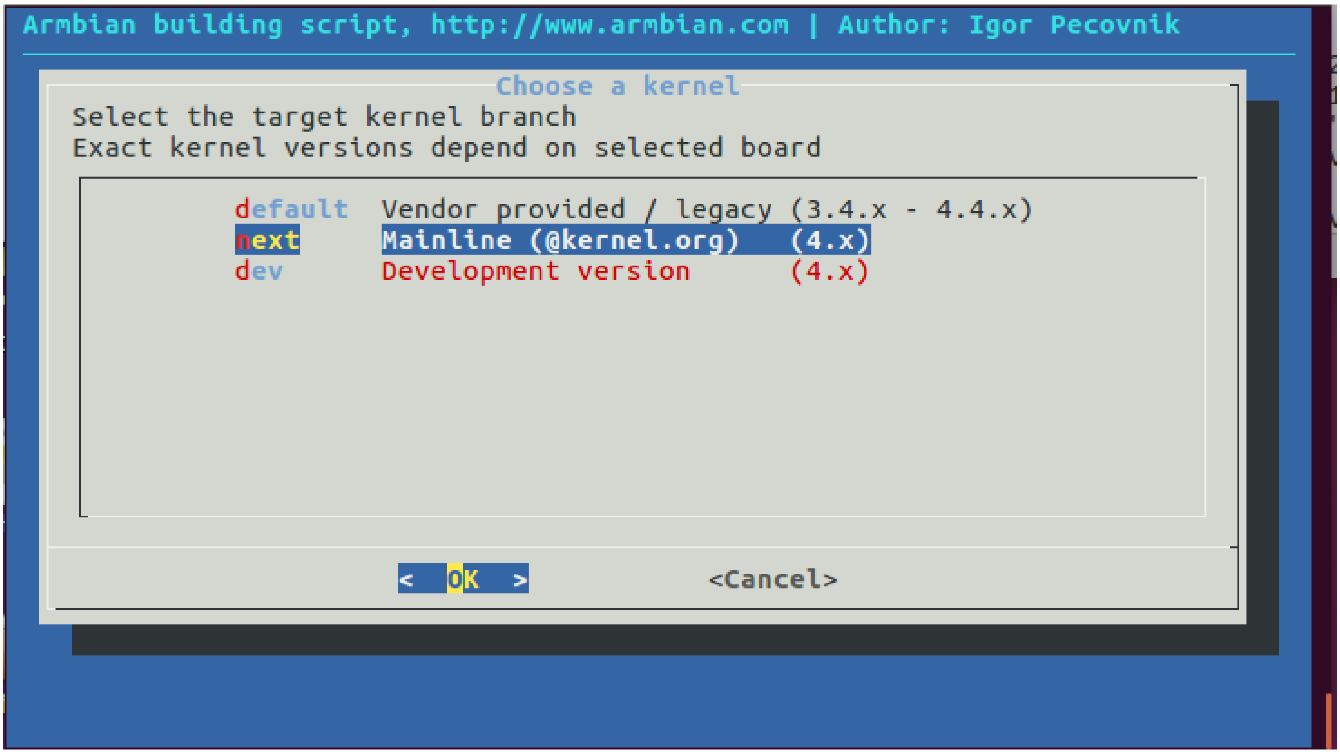


```
Armbian building script, http://www.armbian.com | Author: Igor Pecovnik

                         Choose a board
 Select the target board. Displaying:
  - Officially supported boards
    ↑(-)
 orangepipc             H3 quad core 1GB RAM
 orangepipcplus         H3 quad core 1GB RAM WiFi eMMC
 orangepiplus2e         H3 quad core 2GB RAM WiFi eMMC
 orangepiplus           H3 quad core 1GB/2GB RAM WiFi eMMC
 orangepiprime          H5 quad core 2GB RAM Wi-Fi/BT
 orangepi-r1            H2+ quad core 256MB Wi-Fi + dual Ethernet
 orangepiwin            A64 quad core 1GB SoC GBE WiFi
 orangepizero           H2+ quad core 256/512MB SoC Wi-Fi/Ethernet
 orangepizeroplus2-h3   H3 quad core 512MB SoC Wi-Fi/BT
 orangepizeroplus2-h5   H5 quad core 512MB SoC Wi-Fi/BT
    ↓(+)                                                  87%

     <       OK       >   <Show CSC/WIP/EOS>  <      Cancel      >
```

«Mainline» kernel branch:



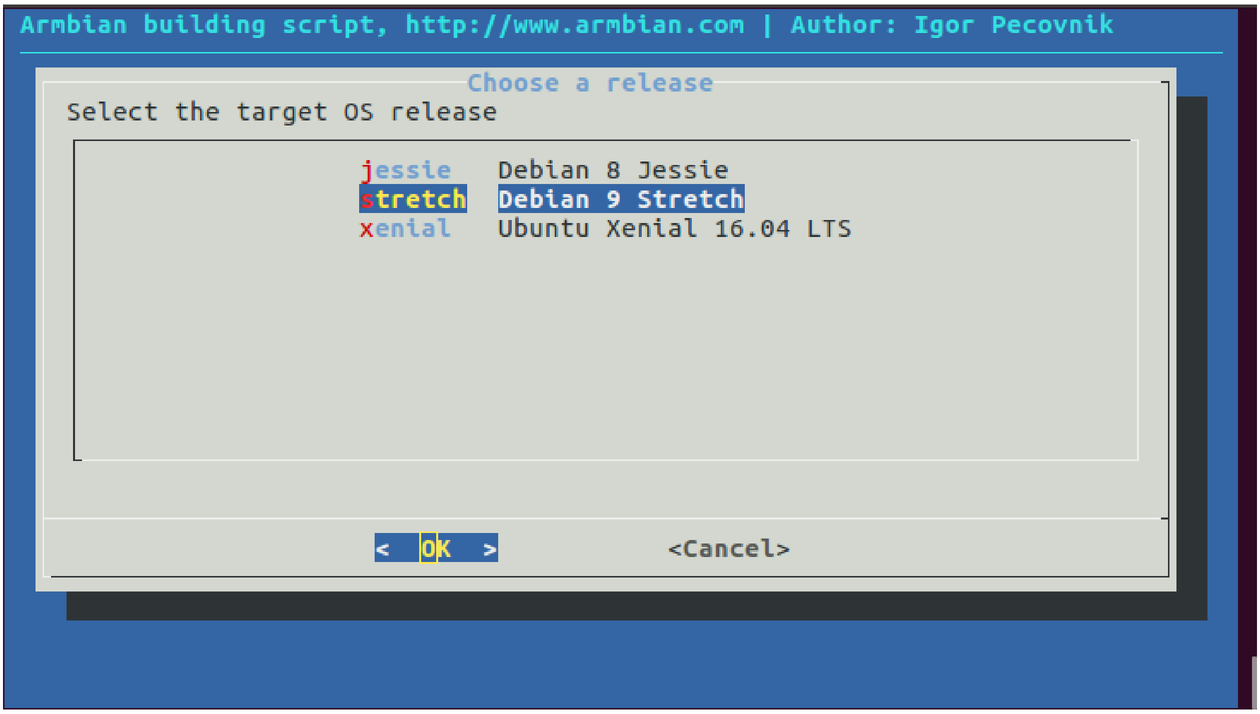
```
Armbian building script, http://www.armbian.com | Author: Igor Pecovnik

                         Choose a kernel
 Select the target kernel branch
 Exact kernel versions depend on selected board

           default   Vendor provided / legacy (3.4.x - 4.4.x)
           next      Mainline (@kernel.org)    (4.x)
           dev       Development version       (4.x)

                  <  OK  >           <Cancel>
```

«`stretch`» release:



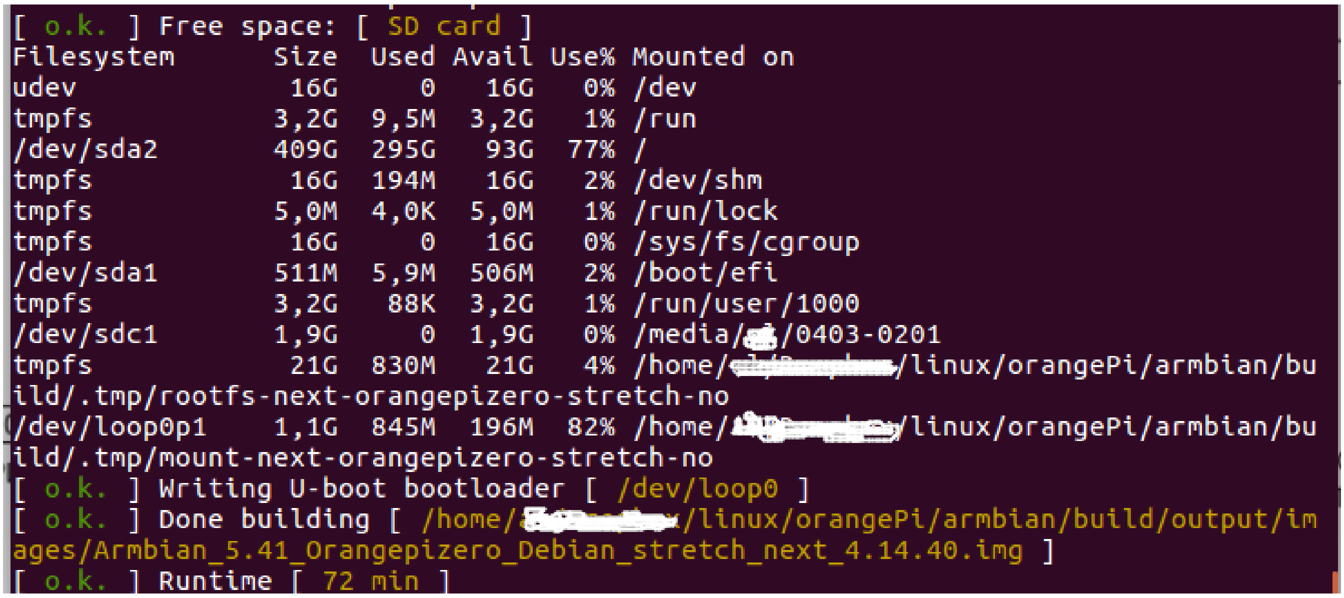
```
Armbian building script, http://www.armbian.com | Author: Igor Pecovnik

Choose a release
Select the target OS release

      jessie    Debian 8 Jessie
      stretch   Debian 9 Stretch
      xenial    Ubuntu Xenial 16.04 LTS

       <  OK  >            <Cancel>
```

Then compilation process will be started. You need to wait while it will be finished. At the end of this process you should see following:



```
[ o.k. ] Free space: [ SD card ]
Filesystem      Size  Used Avail Use% Mounted on
udev             16G     0   16G   0% /dev
tmpfs           3,2G  9,5M  3,2G   1% /run
/dev/sda2       409G  295G   93G  77% /
tmpfs            16G  194M   16G   2% /dev/shm
tmpfs           5,0M  4,0K  5,0M   1% /run/lock
tmpfs            16G     0   16G   0% /sys/fs/cgroup
/dev/sda1       511M  5,9M  506M   2% /boot/efi
tmpfs           3,2G   88K  3,2G   1% /run/user/1000
/dev/sdc1       1,9G     0  1,9G   0% /media/[redacted]/0403-0201
tmpfs            21G  830M   21G   4% /home/[redacted]/linux/orangePi/armbian/bu
ild/.tmp/rootfs-next-orangepizero-stretch-no
/dev/loop0p1    1,1G  845M  196M  82% /home/[redacted]/linux/orangePi/armbian/bu
ild/.tmp/mount-next-orangepizero-stretch-no
[ o.k. ] Writing U-boot bootloader [ /dev/loop0 ]
[ o.k. ] Done building [ /home/[redacted]/linux/orangePi/armbian/build/output/im
ages/Armbian_5.41_Orangepizero_Debian_stretch_next_4.14.40.img ]
[ o.k. ] Runtime [ 72 min ]
```

There is image to write on microSD card.

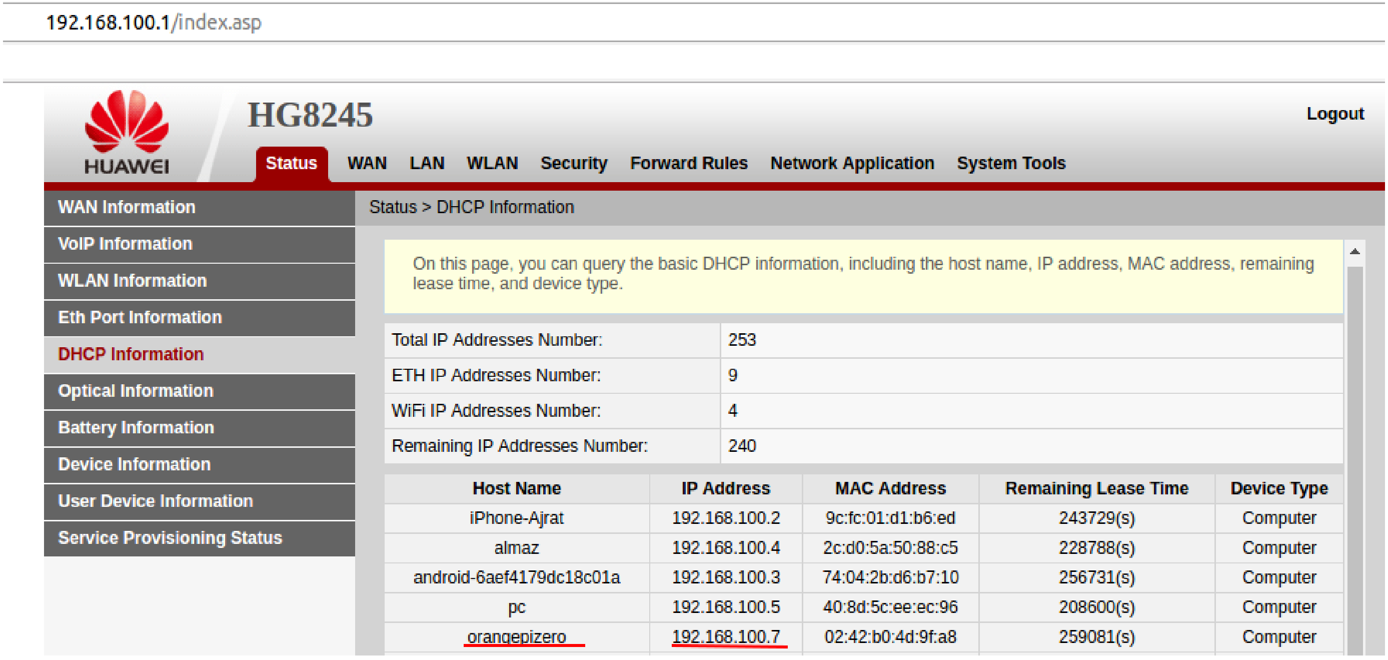
Flashing image in microSD card.

```
sudo dd if=output/images/Armbian_5.41_Orangepizero_Debian_stretch_next_4.14.40.img
of=/dev/sdx
```

Where `/dev/sdx` is your microSD card. Make sure that you flash in SD card.
After that plug microSD card to GW-01 and power up the device.

Remote access via SSH.

Look for `orangepizero` device on your DHCP server:



Status > DHCP Information

On this page, you can query the basic DHCP information, including the host name, IP address, MAC address, remaining lease time, and device type.

Total IP Addresses Number:	253
ETH IP Addresses Number:	9
WiFi IP Addresses Number:	4
Remaining IP Addresses Number:	240

Host Name	IP Address	MAC Address	Remaining Lease Time	Device Type
iPhone-Ajrat	192.168.100.2	9c:fc:01:d1:b6:ed	243729(s)	Computer
almaz	192.168.100.4	2c:d0:5a:50:88:c5	228788(s)	Computer
android-6aef4179dc18c01a	192.168.100.3	74:04:2b:d6:b7:10	256731(s)	Computer
pc	192.168.100.5	40:8d:5c:ee:ec:96	208600(s)	Computer
orangepizero	192.168.100.7	02:42:b0:4d:9f:a8	259081(s)	Computer

Now connect to it via SSH:

```
ssh root@192.168.100.7
```

User/pass to login: root/1234.
You will be asked to change password and add new user.
After that we ready to configure spi interface and install required lorawan packets.

Enabling SPI on armbian on first booting

Check that spidev not exist in list of devices:

```
ls /dev/spidev*
```

Add params to config file to enable SPI interface:

```
nano /boot/armbianEnv.txt
```

Change `overlays` param to:

```
overlays=usbhost2 usbhost3 spi-spidev spi-add-cs1
```

Add following at the end of file:

```
param_spidev_spi_bus=1
param_spidev_spi_cs=1
```

Exit with file saving.
Configure `dts` tree:

```
cp /boot/dtb/overlay/sun8i-h3-spi-spidev.dtbo ~
cd ~
dtc -I dtb -O dts -o sun8i-h3-spi-spidev.dts ./sun8i-h3-spi-spidev.dtbo
nano sun8i-h3-spi-spidev.dts
```

Set folowing in `fragment@2`:

```
status = «ok»;
reg = <0x1>;
```

Exit with file saving.
Then compile new dtbo file and replace old one with that:

```
rm sun8i-h3-spi-spidev.dtbo
dtc -I dts -O dtb -o sun8i-h3-spi-spidev.dtbo ./sun8i-h3-spi-spidev.dts
sudo rm /boot/dtb/overlay/sun8i-h3-spi-spidev.dtbo
sudo cp sun8i-h3-spi-spidev.dtbo /boot/dtb/overlay/
```

After that reboot:

```
reboot
```

After reboot connect to the gateway via ssh as user you created previously.

```
ssh al@192.168.100.7
```

Now you should see spidev1.1 in list of your devices:

```
ls /dev/spidev*
```

SX1301 reset script.

Create SX1301 reset script:

```
cd ~
touch iC880-SPI_reset.sh
nano iC880-SPI_reset.sh
```

Enter following to that file and save it:

```
#!/bin/bash
echo "11" > /sys/class/gpio/export
sleep 2
echo "out" > /sys/class/gpio/gpio11/direction
echo "0" > /sys/class/gpio/gpio11/value
sleep 1
echo "1" > /sys/class/gpio/gpio11/value
sleep 1
echo "0" > /sys/class/gpio/gpio11/value
sleep 1
echo "2" > /sys/class/gpio/export
sleep 2
echo "out" > /sys/class/gpio/gpio1/direction
echo "1" > /sys/class/gpio/gpio1/value
sleep 5
echo "0" > /sys/class/gpio/gpio1/value
sleep 1
echo "0" > /sys/class/gpio/gpio1/value
```

Then set is as executable:

```
sudo chmod +x iC880-SPI_reset.sh
```

Installing lora_gateway.

Download and modify the library:

```
cd ~
git clone https://github.com/Lora-net/lora_gateway.git
cd lora_gateway
nano libloragw/src/loragw_spi.native.c
```

Change `/dev/spidev0.0` to `/dev/spidev1.1` and save file.
Complie the library:

```
make
```

Installing packet_forwarder.

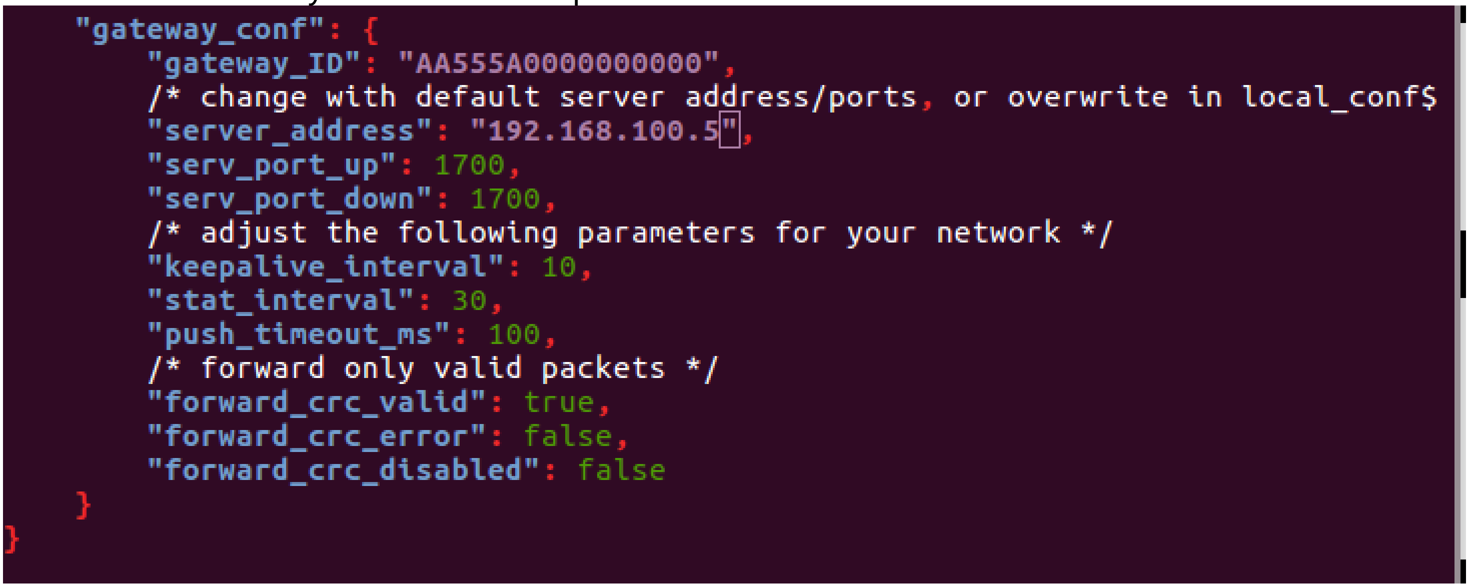
Download and compile the library:

```
cd ~
git clone https://github.com/Lora-net/packet_forwarder.git
cd packet_forwarder/
./compile.sh
```

Set lora server IP address:

```
cd lora_pkt_fwd/cfg/
cp global_conf.json.PCB_E286.EU868.basic ../global_conf.json
nano global_conf.json
```

Set «server_port_up» and «server_port_down» to port your lora server uses. Set «server_address» to IP address of lora server you use. For example:



```
    "gateway_conf": {
        "gateway_ID": "AA555A0000000000",
        /* change with default server address/ports, or overwrite in local_conf$
        "server_address": "192.168.100.5",
        "serv_port_up": 1700,
        "serv_port_down": 1700,
        /* adjust the following parameters for your network */
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false
    }
}
```

Now all ready to run the gateway.

Running the gateway.

Run this at every power up of gateway:

```
cd ~
sudo ./iC880-SPI_reset.sh
sudo chmod 777 /dev/spidev1.1
cd packet_forwarder/lora_pkt_fwd
./lora_pkt_fwd
```