LAN865x 10BASE-T1S MAC-PHY Ethernet Linux Driver

This document describes the procedure for configuring the LAN865x hardware and installing driver. These procedures are tested in **Raspberry Pi 4 with Linux Kernel 6.1.21**.

Setup the hardware

- Connect the Pi 4 Click Shield board on the 40-pin header in Pi 4.
  - Pi 4 Click Shield buy link: https://www.mikroe.com/pi-4-click-shield
- Connect the LAN865x on the Mikro Bus slots available on the Pi 4 Click Shield.
  - LAN865x Click board buy link: https://www.mikroe.com/two-wire-eth-click
  - For more details please refer to the Microchip LAN865x product website:
    - LAN8650
    - LAN8651
- Prepare the SD card for Pi 4 with raspian image available in the Pi official website.
  - Link to refer: https://www.raspberrypi.com/software/

Prerequisites

If the Raspberry Pi OS is freshly installed, then the below prerequisites are mandatory before using the driver package. Please make sure the Raspberry Pi is connected with internet because some dependencies need to be installed for driver package installation script.

- Please make sure the **current date and time** of Raspberry Pi is up to date. Can be checked with the below command,

```
 $ date
```

- If they are not up to date and if you want to set it manually then use the below example and modify the fields for the current date and time,

```
 $ sudo date -s "Wednesday 01 March 2023 02:38:43 PM IST"
```

Link to refer: https://raspberrytips.com/set-date-time-raspberry-pi/

- Please make sure the **apt-get is up to date**. Run the below command to update apt-get,

```
 $ sudo apt-get update
```

Get project files

- Extract the downloaded software package into your local directory using the below command,

```
 $ unzip lan865x-linux-driver-0v2.zip
$ cd lan865x-linux-driver-0v2/
```

Configure and install using script

There are two scripts are available for the user interface. You can use either one of the script to configure LAN865x. Run the below command if you wan to go with command line interface.

```
 $ ./prepare_lan865x_cli.sh
```

Or run the below command if you want to go with terminal user interface.

```
 $ ./prepare_lan865x_tui.sh
```

Note: If the above script is failed to run because of the executable permission then try after running the below commands,

```
 $ chmod +x prepare_lan865x_cli.sh
$ chmod +x prepare_lan865x_tui.sh
```

- The script checks for the **dialog**, **device-tree-compiler** and **raspberrypi-kernel-headers** packages installation status and asks for your confirmation to install in case if they are not installed.
- The script provides the following three options to prepare the LAN865x in Raspberry Pi 4.
  - **Configure 1st LAN865x (eth1)** - Provides options for configuring PLCA settings, MAC address, IP address and Save & Configure for the 1st LAN865x connected in the Mikroe bus 1 slot. **Note:** Always do **Save & Configure** in case if you modified something to take effect otherwise your changes will be lost.
  - **Configure 2nd LAN865x (eth2)** - Provides options for configuring PLCA settings, MAC address, IP address and Save & Configure for the 2nd LAN865x connected in the Mikroe bus 2 slot. **Note:** Always do **Save & Configure** in case if you modified something to take effect otherwise your changes will be lost.

- **Install Driver** - Compiles and installs the driver.
- In a fresh installation, **Save & Configure** and **Install Driver** are must even if you want to go with the default values and then **Reboot** the Raspberry Pi.
- Please explore each and every option to get to know about the usage.
- A **reboot** is always required in case you configured any of the above settings to take effect.

Note: IP address assignment is done based on the MAC address configuration. An invalid MAC address configuration may lead to a random IP assignment.

Configure manually

- Open **config.txt** file which is located into the **/boot/** directory with **Superuser** access to include **lan865x device tree overlay**.
    - Uncomment the line **#dtparam=spi=on**
    - Add the line **dtoverlay=lan865x** after the above line, so the **config.txt** should have the below lines also,

Command to open the file,

```
$ sudo vim /boot/config.txt
```

Contents to add in the file,

```
 dtparam=spi=on
dtoverlay=lan865x
```

Note: A sample **config.txt** file with above settings is available in the **config** directory for the reference.

Configure and compile device tree overlay,

- The above hardware setup can support the maximum of two LAN865x 10BASE-T1S Ethernet devices and they can be configured in the device tree overlay file **lan865x-overlay.dts** which is located in the **dts** directory. Valid value ranges are commented nearby each property in the **lan865x-overlay.dts** file.
- Edit the desired properties values in the device tree such as,
    - local-mac-address
        - Enter the desired MAC address
    - plca-enable
        - 1 - PLCA enable and CSMA/CD disable, 0 - vice versa.
    - plca-node-id
        - 0 to 254
    - plca-node-count
        - 1 to 255
    - plca-burst-count
        - 0x0 to 0xFF
    - plca-burst-timer
        - 0x0 to 0xFF
    - plca-To-timer
        - 0x0 to 0xFF
    - tx-cut-through-mode
        - 1 - Tx cut through mode enable and store & forward mode disable, 0 - vice versa.
    - rx-cut-through-mode
        - 1 - Rx cut through mode enable and store & forward mode disable, 0 - vice versa.
    - oa-chunk-size
        - 32 or 64.
    - oa-protected
        - 1 - OA protected enable, 0 - OA protected disable.

Command to open the file,

```
$ vim dts/lan865x-overlay.dts
```

Note: Tx and Rx cut through mode will fail if SPI transfer rate is slower than the network transfer rate.

- Make sure the device tree compiler **dtc** is installed in Pi, if not use the below command to install it,

```
$ sudo apt-get install device-tree-compiler
```

- Compile the device tree overlay file **lan865x-overlay.dts** to generate **lan865x.dtbo** using the below command,

```
$ dtc -I dts -O dtb -o lan865x.dtbo dts/lan865x-overlay.dts
```

- Copy the generated **lan865x.dtbo** file into the **/boot/overlays/** directory using the below command,

```
$ sudo cp lan865x.dtbo /boot/overlays/
```

- Now reboot the Pi.

- Steps to compile and load the driver module

- Make sure the **linux headers** are installed in Pi, if not use the below command to install it,

```
$ sudo apt-get --assume-yes install build-essential cmake subversion libncurses5-dev bc bison flex libssl-dev python2
$ sudo wget https://raw.githubusercontent.com/RPi-Distro/rpi-source/master/rpi-source -O /usr/local/bin/rpi-source && sudo chmod
$ rpi-source --skip-gcc
```

- Compile the driver using the below command,

```
$ cd lan865x-linux-driver/
$ make
```

- Load the driver using the below command,

```
$ sudo insmod lan865x_t1s.ko
```

- Now you are ready with your 10BASE-T1S ethernet interfaces **eth1** and **eth2**.
- Use the below command to enable better performance in Pi,

```
$ echo performance | sudo tee /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor > /dev/null
```

Note: The above setting needs to be done after every boot. **Example ethernet configuration**

```
$ sudo ip addr add dev eth1 192.168.10.11/24
$ sudo ip addr add dev eth2 192.168.20.21/24
```

Tips

- If you don't want to do the above **driver loading**, **setting scaling_governor** and **ip configuration** in every boot then add those commands in the **/etc/rc.local** file so that they will be executed automatically every time when you boot Pi. For that open the **rc.local** file with superuser permission and add the following lines before **exit 0**,

Command to open the file,

```
$ sudo vim /etc/rc.local
```

Contents to add in the file,

```
cd <your_local_directory_for_driver_module>
insmod lan865x_t1s.ko
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
sleep 1
sudo ip addr add dev eth1 192.168.10.11/24
sudo ip addr add dev eth2 192.168.20.21/24
```

OR

- You can add the IP configuration alone in **dhcpcd.conf** file which is located in the **/etc/** directory like below,

Command to open the file,

```
$ vim /etc/dhcpcd.conf
```

Contents to add in the file,

```
interface eth1
static ip_address=192.168.10.11/24
interface eth2
static ip_address=192.168.20.21/24
```

Testing results with iperf3 in RPI 4

- Command on server side and the IP is 192.168.10.12,

```
$ iperf3 -s -i 1
```

- Command on client side and the IP is 192.168.10.11,

```
$ iperf3 -c 192.168.10.12 -b 10M -u -t 86400 -i 1
```

Test case 1: Single LAN865x connected and tx cut through mode enabled.

Result: Performance around **9.4Mbps**.

Test case 2: Single LAN865x connected and tx cut through mode disabled.

Result: Performance around **9.4Mbps**.

Test case 3: Two LAN865x's are connected and tx cut through mode enabled on both.

Result: The total bandwidth configuration should not exceed 10Mbps to achieve the expected performance.

Test case 4: Two LAN865x's are connected and tx cut through mode disabled on both.

Result: Performance around **5.0Mbps** on both when both LAN865x's are configured as Tx. In case of Rx, the total bandwidth configuration should not exceed more than 10 Mbps to achieve the expected performance.

Note 1: Test case 3 and 4 are tested with two networks and also note that the above hardware setup is using **single SPI master** and **two LAN865x SPI slaves**.

Note 2: The above tests are performed in RPI 4. Different platforms with dedicated SPI master for each nodes will give better performance than this.

TODO

- Timestamping according to Open Alliance TC6 is to be implemented.

References

- OPEN Alliance TC6 - 10BASE-T1x MAC-PHY Serial Interface specification
- OPEN Alliance TC6 Protocol Driver for LAN8650/1