How to setup the Raspberry Pi to use the LPRF Driver

1. Download and unzip Raspbian image

$ wget [Link to download]

$ unzip *file* -d *destination*

1. Connect the SD-Card to the PC
2. Get the image to the SD-Card

$ df -h // determine the partition number

$ umount /dev/sdb1

$ umount /dev/sdb2

$ sudo dd bs=4M if=*file.img* of=/dev/sdb

$ pkill -USR1 -n -x dd // to get status

$ sync

1. Insert the SD-Card into the RPI and start the RPI
2. RPI configuration (first start config)

* Expand Filesystem
* Boot Options -> B1
* Change Timezone and Keyboard layout

1. Reboot
2. Change network settings

$ sudo nano /etc/network/interfaces

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet dhsp

hwaddress 02:E4:73:03:BE:83

$ sudo reboot

1. Changing the Host

$ sudo nano /etc/hosts

Change “raspberrypi” into the name you want (RPI2)

$ sudo nano /etc/hostname

Replace “raspberrypi” with the name you choose above

$ sudo /etc/init.d/hostname.sh

$ sudo reboot

1. Now you can use ssh for remote to the RPI

$ ssh [pi@137.226.200.211](mailto:pi@137.226.200.211)

1. Update the RPI

$ sudo apt-get update

$ sudo apt-get upgrade

1. Download the linux source tree

$ mkdir kernel

$ cd kernel

$ git clone –depth=1 <https://github.com/raspberrypi/linux.git>

1. Prepare kernel compiling

$ sudo apt-get install bc libncurses5-dev libncursesw5-dev

$ KERNEL=kernel

$ make bcmrpi\_defconfig

$ make menuconfig

* Device Drivers -> Network device support -> Wireless LAN -> Realtek 8192C USB WiFi
* Device Drivers -> Network device support -> USB Network Adapters -> Multi-purpose USB Networking Framework -> SMSC LAN95XX based USB 2.0 10/100 ethernet devices
* Device Drivers -> SPI support -> BCM2835 SPI controller <\*>
* Device Drivers -> USB support -> USB Mass Storage support <\*>
* Device Drivers -> USB support -> DesignWare USB2 DRD Core Support <\*>
* Networking support -> RF switch subsystem support <\*>
* Networking support -> Wireless -> cfg80211 – wireless configuration API <\*>
* Networking support -> Wireless -> Generic IEEE 802.11 Networking Stack (mac80211) <\*>
* Networking support -> Networking Options -> TCP/IP networking -> The IPv6 protocol <\*>
* Networking support -> Networking Options -> 6LoWPAN Support <\*>
* Networking support -> Networking Options -> IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support <\*>
* Networking support -> Networking Options -> IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support -> IEEE 802.15.4 socket interface <\*>
* Networking support -> Networking Options -> IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support -> 6lowpan support over IEEE 802.15.4 <\*>
* Networking support -> Networking Options -> IEEE Std 802.15.4 Low-Rate Wireless Personal Area Networks support -> Generic IEEE 802.15.4 Soft Networking Stack (mac802154) <\*>
* Networking support -> Networking Options -> NETLINK: mmaped IO <\*>
* Networking support -> Networking Options -> NETLINK: socket monitoring interface <\*>
* Device Drivers -> Network Device Support -> IEEE 802.15.4 drivers -> AT86RF230/231/233/212 transceiver driver <M>

1. Compiling the kernel

$ make zImage modules dtbs

Takes almost 12 hours

1. Install Modules and copy files in the boot folder

$ sudo make modules\_install

$ sudo cp arch/arm/boot/dts/\*.dtb /boot/

$ sudo cp arch/arm/boot/dts/overlays/\*.dtb /boot/overlays

$ sudo cp arch/arm/boot/dts/overlays/README /boot/obverlays

$ sudo scripts/mkknlimg arch/arm/boot/zImage /boot/kernel.img

1. Modify /boot/config.txt

Insert at the end of the file

dtoverlay=mmc

1. Reboot
2. Modify arch/arm/boot/dts/bcm2708-rpi-b-plus.dts

Delete the old spi0-node and insert

&spi0 {

status = "okay";

at86rf231@0 {

compatible = "atmel,at86rf231";

reg = <0>;

interrupts = <23 1>;

interrupt-parent = <&gpio>;

reset-gpio = <&gpio 24 1>;

sleep-tpio = <&gpio 25 1>;

spi-max-frequency = <500000>;

};

};