How to enable SPI under BananaPi BPI-R4 v3.

1. Intro

All necessary files are located in my repo: https://github.com/drozdi70/bananapi_bpir4
Especially Adafruit_Blinka_SPI_Pack.tar used in point 5.

2. Image preparation.

Add in DTS file section related to enabling spi/spidev (~/openwrt/target/linux/mediatek/files-6.6/arch/arm64/boot/dts/mediatek/mt7988a-bananapi-bpi-r4.dts):

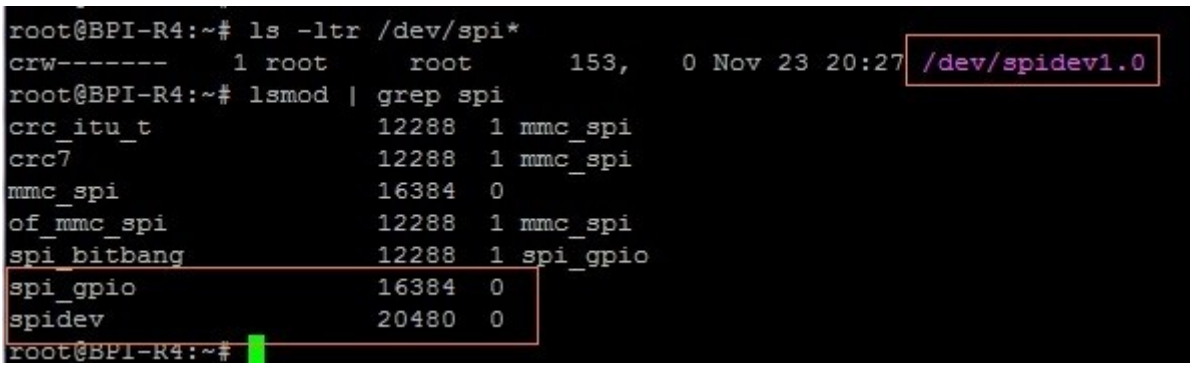
```
&spi1 {
	#address-cells = <1>;
	#size-cells = <0>;
	pinctrl-names = "default";
	pinctrl-0 = <&spi1_pins>;
	status = "okay";

	spidev0: spidev@0 {
		compatible = "sitronix,st7789v";
		spi-max-frequency = <32000000>;
		#address-cells = <1>;
		#size-cells = <0>;
		reg = <0>;
		dc-gpios = <&pio 50 GPIO_ACTIVE_HIGH>;    //GPIO50/pin 15
		reset-gpios = <&pio 53 GPIO_ACTIVE_LOW>;  //pin 22 / GPIO53
		spi-cpol;
		spi-cpha;
		status = "okay";
	};
};
```

Install a patch for st7789v – 0004-spidev-lcd-6.6.patch in your directory where you compile openwrt:
cp 0004-spidev-lcd-6.6.patch ~/openwrt/target/linux/mediatek/patches-6.6/

In menuconfig → enable spi/spipdev to be sure all is installed as
kmod-spi-dev spi_tools spidev_test kmod-spi-gpio

Write image to SD and start router from SD card.
Make connection to internet working from your router.

```
# apk update
# apk upgrade
```



3. SPI bus check

If nothing is connected to the bus:

root@BPI-R4:~# spidev_test -D /dev/spidev1.0 -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D |......@.....|
RX | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|

Next we short pin19 and pin21:

root@BPI-R4:~# spidev_test -D /dev/spidev1.0 -v
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 kHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D |......@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0D |......@.....|
root@BPI-R4:~#

root@BPI-R4:~# spi-config -d /dev/spidev1.0 -q
/dev/spidev1.0: mode=0, lsb=0, bits=8, speed=32000000, spiready=0

4. ST7899 LCD screen wiring to the board

BPI-R4 GPIO pinout:
https://docs.banana-pi.org/en/BPI-R4/GettingStarted_BPI-R4#_gpio_define

SPI - 1.69inch LCD Module 240x280 pixels, ST7789V2
SPI - 1.54inch LCD module 240x240 pixels, ST7789

VCC -> 3.3V (pin 17)
GND -> GND (pin 20)
SCL -> SPI1_CLK (pin 23) GPIO31
SDA -> SPI1_MOSI (pin 19) GPIO30
RST -> pin 22 / GPIO53
DC -> pin 15 / GPIO50
CS -> pin26 / GPIO52
BL -> +VCC or not connected

SPI1_CSB (24) GPIO28 -- not connected!
SPI1_MISO (21) GPIO29 -- not connected!

How to check which GPIO line is free and can be used?

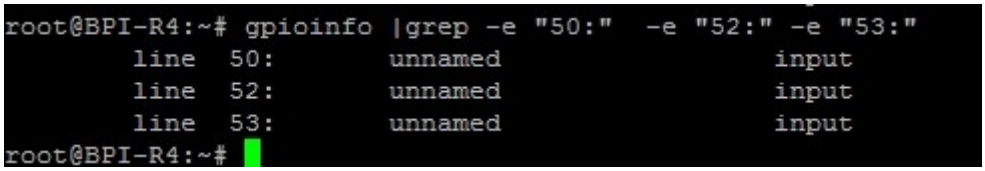
root@BPI-R4:~# apk add gpiod-tools

root@BPI-R4:~# gpioinfo
gpiochip0 - 84 lines:
line 0: unnamed output consumer=tx-disable
line 1: unnamed input consumer=tx-fault
line 2: unnamed input consumer=los
line 3: unnamed input active-low consumer=rate-select0
line 4: unnamed input
line 5: unnamed output active-low consumer=reset
line 6: unnamed input
line 7: unnamed input
line 8: unnamed input
line 9: unnamed input consumer=kernel
line 10: unnamed input consumer=kernel
line 11: unnamed input
line 12: unnamed input active-low consumer=cd
line 13: unnamed input
line 14: unnamed input active-low consumer=WPS
line 15: unnamed input consumer=kernel
line 16: unnamed input consumer=kernel
line 17: unnamed input
line 18: unnamed input
line 19: unnamed input
line 20: unnamed input consumer=kernel
line 21: unnamed input active-low consumer=rate-select0
line 22: unnamed input consumer=kernel

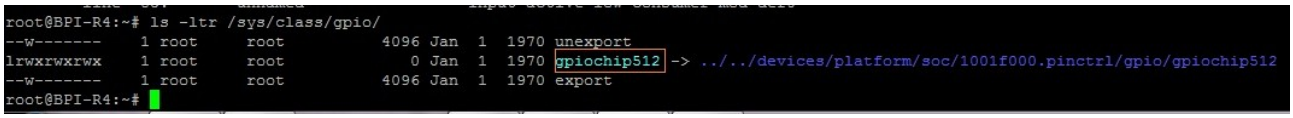
line 23: unnamed input consumer=kernel
line 24: unnamed input consumer=kernel
line 25: unnamed input consumer=kernel
line 26: unnamed input consumer=kernel
line 27: unnamed input consumer=kernel
line 28: unnamed input consumer=kernel
line 29: unnamed input consumer=kernel
line 30: unnamed input consumer=kernel
line 31: unnamed input consumer=kernel
line 32: unnamed input consumer=kernel
line 33: unnamed input consumer=kernel
line 34: unnamed input consumer=kernel
line 35: unnamed input consumer=kernel
line 36: unnamed input consumer=kernel
line 37: unnamed input consumer=kernel
line 38: unnamed input
line 39: unnamed input
line 40: unnamed input
line 41: unnamed input
line 42: unnamed input
line 43: unnamed input
line 44: unnamed input
line 45: unnamed input
line 46: unnamed input
line 47: unnamed input
line 48: unnamed input
line 49: unnamed input
line 50: unnamed input
line 51: unnamed input
line 52: unnamed input
line 53: unnamed input
line 54: unnamed input consumer=los
line 55: unnamed input consumer=kernel
line 56: unnamed input consumer=kernel
line 57: unnamed input consumer=kernel
line 58: unnamed input consumer=kernel
line 59: unnamed input consumer=kernel
line 60: unnamed input consumer=kernel
line 61: unnamed input consumer=kernel
line 62: unnamed input
line 63: unnamed output consumer=blue:wps
line 64: unnamed input consumer=kernel
line 65: unnamed input consumer=kernel
line 66: unnamed input consumer=kernel
line 67: unnamed input consumer=kernel
line 68: unnamed input
line 69: unnamed input consumer=tx-fault
line 70: unnamed output consumer=tx-disable
line 71: unnamed input consumer=kernel
line 72: unnamed input consumer=kernel
line 73: unnamed input
line 74: unnamed input consumer=kernel

```
line  75:      unnamed         input
line  76:      unnamed         input
line  77:      unnamed         input consumer=kernel
line  78:      unnamed         input consumer=kernel
line  79:      unnamed         output consumer=green:status
line  80:      unnamed         input consumer=kernel
line  81:      unnamed         input consumer=kernel
line  82:      unnamed         input active-low consumer=mod-def0
line  83:      unnamed         input active-low consumer=mod-def0
```

All lines consumer=kernel cannot be used as already reserved.
We check lines (GPIO) 50,52,53



We also check base numer for gpiochip → in our case is 512 (needed for pin mapping).



5. SPI configuration

SPI can work programmed in different ways, I decided to use Adafruit-Blinka package and adopt it a little bit as per now our board is not supported offcially there.
Please remember one fact that in the case of any possible upgrade of the package Adafruit_Blinka you need to repeat whole process as probably some files will be replaced by new ones from new release.

Adafruit-Blinka version: 8.50.0
Adafruit-platformdetect version: 3.75.0

apk update
apk upgrade
apk add python3 curl git git-http
apk add python3-dev make
apk add python3-setuptools
apk add python3-pip sudo coreutils
apk add python3-pillow
apk add python3-gpiod

pip3 install Adafruit-Blinka

pip3 install adafruit-platformdetect
pip3 install --upgrade adafruit-python-shell click
pip3 install adafruit-circuitpython-st7789
pip install adafruit-circuitpython-display-text
pip install Adafruit-GPIO

Some Hocus-Pocus (based on https://github.com/Dangku/Adafruit_Python_PlatformDetect)

cd /usr/lib/python3.11/site-packages
mv adafruit_platformdetect adafruit_platformdetect.ORIG
copy adafruit_platformdetect.tar here and extract
tar xvf adafruit_platformdetect.tar
mv board.py board.py.ORIG
copy board.py here
copy bananapi.tar to /usr/lib/python3.11/site-packages/adafruit_blinka/board
mv bananapi bananapi.ORIG
tar xvf bananapi.tar
copy mt7988a.tar to /usr/lib/python3.11/site-packages/adafruit_blinka/microcontroller
tar xvf mt7988a.tar
in /usr/lib/python3.11/site-packages/
mv digitalio.py digitalio.py.ORIG
in /usr/lib/python3.11/site-packages/
mv microcontroller microcontroller.ORIG
copy microcontroller.tar here and extract
tar xvf microcontroller.tar

Tests:

root@BPI-R4:~# python3 detect.py
Board Detection Test

Check that the Chip and Board IDs match your board and that this it is
correctly detecting whether or not it is a Linux board.

Board deteted:

Chip id: MT7988A
Board id: BANANA_PI_BPI_R4

Linux Detection

Is this an embedded Linux system? True

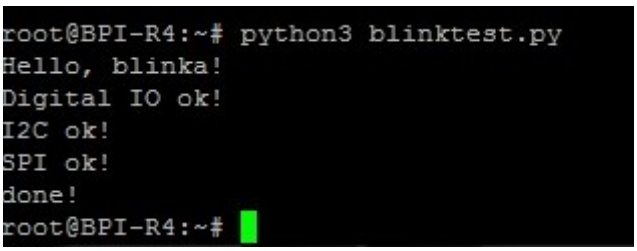
Raspberry Pi Boards

Is this a Pi 3B+? False
Is this a Pi 4B? False
Is this a 40-pin Raspberry Pi? False
Is this a Raspberry Pi Compute Module? False

Other Boards

Is this a Siemens Simatic IOT2000 Gateway? False
Is this a BananaPi board? True
Is this a 96boards board? False
Is this a BeagleBone board? False
Is this a Giant board? False
Is this a Coral Dev board? False
Is this a MaaXBoard? False
Is this a SiFive board? False
Is this a PYNQ board? False
Is this a Rock Pi board? False
Is this a NanoPi board? False
Is this a Khadas VIM3 board? False
Is this a Clockwork Pi board? False
Is this a Seeed Board? False
Is this a UDOO board? False
Is this an ASUS Tinker board? False
Is this an STM32MP1 board? False
Is this a MilkV board? False
Is this a Luckfox Pico board? False
Is this a generic Linux PC? False
Is this an OS environment variable special case? False

BananaPi board detected.



Remark:
Adafruit-Blinka package could be probably pinned / hold to the current version/state as below by runnig:

```
# pip3 freeze > requirements.txt
```

and revert operation:

```
# pip3 install -r requirements.txt
```

Ref. https://builtin.com/software-engineering-perspectives/pip-freeze

6. Examples for LCD ST7899

git clone https://github.com/adafruit/Adafruit_CircuitPython_ST7789

cd Adafruit_CircuitPython_ST7789/examples/

edit a file and adjust SPI lines parameteres as below:

spi = busio.SPI(board.D23, MOSI=board.D19)
tft_cs = board.D26
tft_dc = board.D15
tft_backlight = None

or

tft_dc = board.D15
tft_cs = board.D26
spi_clk = board.D23
spi_mosi = board.D19
tft_rst = board.D22
backlight = None
spi = busio.SPI(spi_clk, spi_mosi)

or

spi = board.SPI()
tft_cs = board.D26
tft_dc = board.D15

display_bus = FourWire(spi, command=tft_dc, chip_select=tft_cs, reset=board.D22)

7. XPT2046 touchscreen wiring to the board

TFT	Board	GPIO	Pin #
T_CLK	SPI1_CLK	GPIO31	23
T_CS		GPIO52	26
T_DIN	SPI1_MOSI	GPIO30	19
T_DO	SPI1_MISO	GPIO29	21
T_IRQ		GPIO50	15
T_RST		GPIO53	22

VCC -> 5V (pin 2 or pin 4)
GND -> GND (pin 20)

Installation:

```
# pip3 install xpt2046-circuitpython
```
pip3 install adafruit-circuitpython-rgb-display

8. Examples for touchscreen XPT2046

```
git clone https://github.com/humeman/xpt2046-circuitpython
```

cd xpt2046-circuitpython/sample

Setting proper values for GPIO lines as:

from board import D23, D19, D21, D26, D15

# Pin config
T_CS_PIN = D26
T_IRQ_PIN = D15
MOSI = D19
SCK = D23
MISO = D21

In my case touchscreen is not returning correct value ;)

9. Referrences/Docs

https://docs.circuitpython.org/projects/st7789/en/stable/examples.html
https://github.com/russhughes/st7789_mpy/tree/master
https://forum.banana-pi.org/t/banana-pi-bpi-r64-spi-touch-panel-test-with-openwrt/10009/2
https://www.coderdojotc.org/micropython/displays/graph/14-lcd-st7789V/
https://github.com/rm-hull/luma.examples
https://forum.banana-pi.org/t/bpi-r2-r3-and-ssd1306-oled-screen/11917/47
https://github.com/abhra0897/stm32f1_st7789_spi
https://git.datalabrotterdam.nl/customer/projects/Micropython-examples/-/tree/main/SPI%20ST7789%20Display?ref_type=heads
https://github.com/solinnovay/Python_ST7789
https://github.com/sonocotta/st7789-orangepi-python

https://techatronic.com/st7789-display-pi-pico/#google_vignette
https://forum.banana-pi.org/t/spi-touchscreen-ads7846-xpt2046-on-ubuntu-images-bpi-r2/4781/29
https://blog.embeddedexpert.io/?p=1215
https://github.com/pimoroni/st7789-python
https://techatronic.com/st7789-raspberry-pi/
https://github.com/pimoroni/st7789-python
https://github.com/devbis/st7789_mpy
https://forums.pimoroni.com/t/adafruit-st7789-1-54-python-code/14238
https://git.datalabrotterdam.nl/customer/projects/Micropython-examples/-/tree/main/SPI%20ST7789%20Display
https://github.com/russhughes/ttgo-hershey-fonts
https://github.com/solinnovay/Python_ST7789
https://raspberrypi.stackexchange.com/questions/104846/how-to-connect-st7789-lcd-to-spi-bus-1-as-2nd-screen
http://helloraspberrypi.blogspot.com/2021/02/raspberry-pi-picomicropython-st7789-spi.html
https://python.scitoys.com/st7789
https://www.programcreek.com/python/example/101401/machine.SPI
https://pypi.org/project/xpt2046-circuitpython/