

# Raport z laboratorium 1 - 6.03.2024

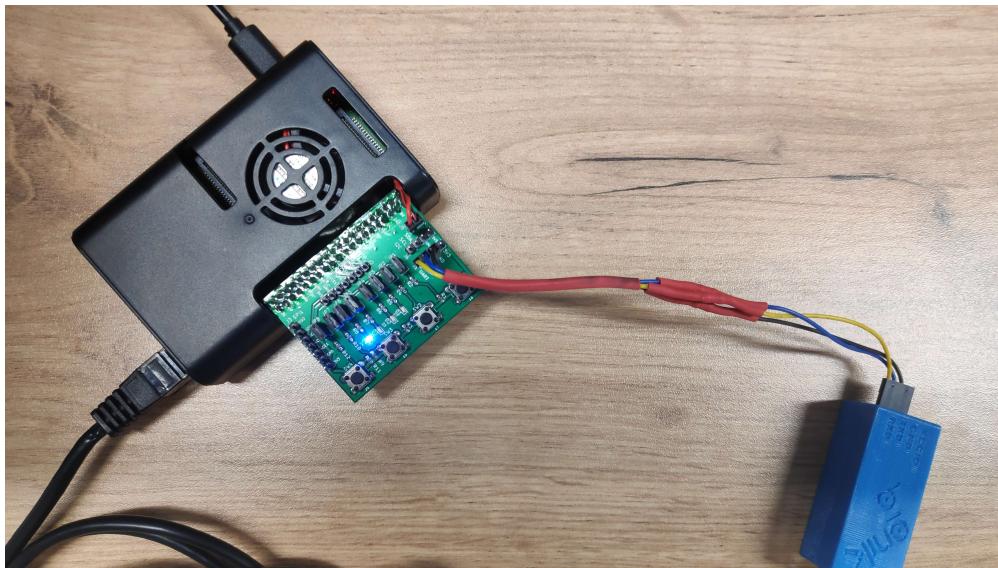
Sebastian Abramowski, 325142

Bogumił Stoma, 325233

Raport wygenerowany automatycznie z pliku rapport.md

## Przygotowanie stanowiska

Zrobiliśmy według instrukcji



## Pierwsze uruchomienie RPi

Podłączliśmy się do terminala UART do RPi przez program tio

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
user@lab-44:~$ tio /dev/ttyUSB0
[tio 12:53:16] tio v1.29
[tio 12:53:16] Press ctrl-t q to quit
[tio 12:53:16] Connected
[ 35.805061] cam-dummy-reg: disabling
[ 35.808673] cam1-reg: disabling

Welcome to Buildroot rescue os
rescue login: root
# ls
adapter_test.sh
#
```

Potem uruchomiliśmy DHCP, aby RPi dostało adres IP

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
user@lab-44:~$ tio /dev/ttyUSB0
[tio 12:53:16] tio v1.29
[tio 12:53:16] Press ctrl-t q to quit
[tio 12:53:16] Connected
[ 35.805061] cam-dummy-reg: disabling
[ 35.808673] cam1-reg: disabling

Welcome to Buildroot rescue OS
rescue login: root
# ls
adapter_test.sh
# udhcpc
# udhcpc: started, v1.33.1
udhcpc: sending discover
udhcpc: sending select for 10.42.0.63
udhcpc: lease of 10.42.0.63 obtained, lease time 3600
deleting routers
adding dns 10.42.0.1
# 
```

Następnie odczytaliśmy IP RPi i hosta oraz sprawdziliśmy czy pingowanie działa w obie strony

Adres RPi: 10.42.0.63

Adres hosta: 10.42.0.1

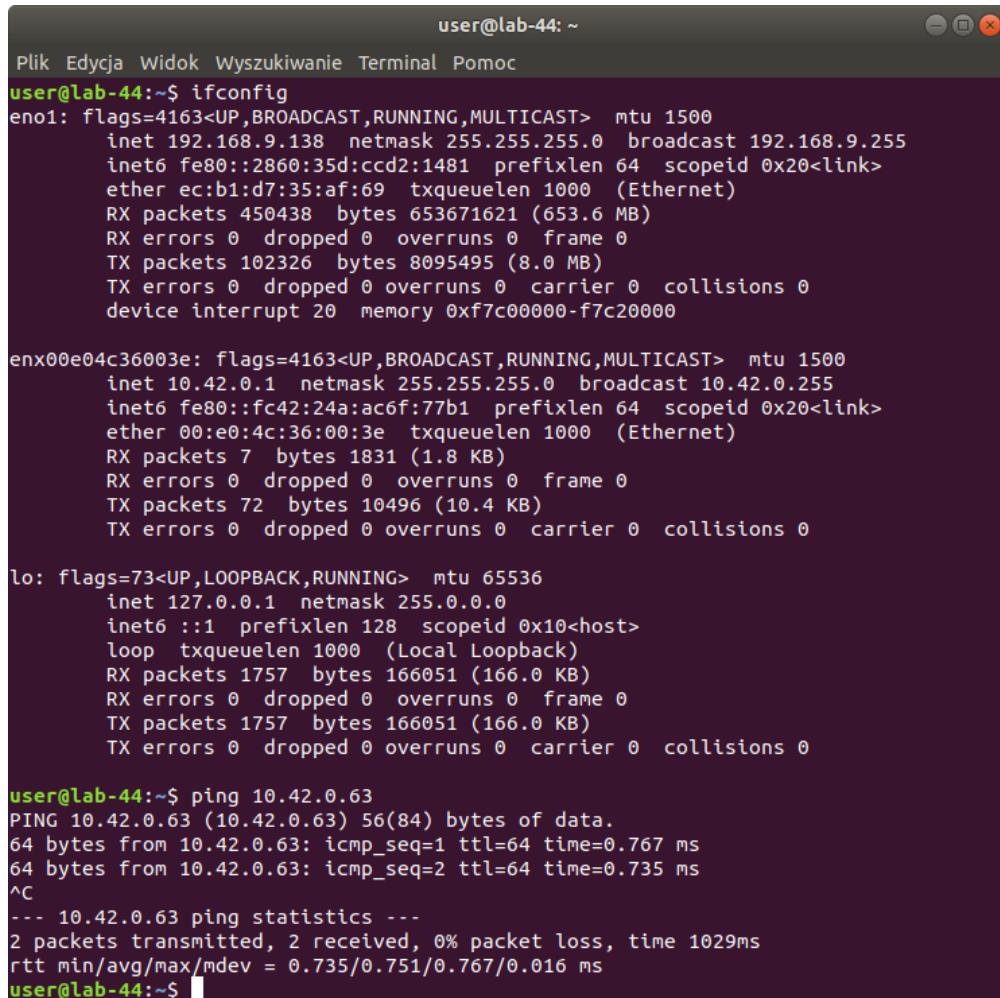
Na RPi

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
ip tunnel add|change|del|show [NAME]
# ifconfig
eth0      Link encap:Ethernet HWaddr E4:5F:01:2B:41:65
          inet addr:10.42.0.63 Bcast:10.42.0.255 Mask:255.255.255.0
          inet6 addr: fe80::e65f:1ff:fe2b:4165/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11 errors:0 dropped:0 overruns:0 frame:0
            TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:2072 (2.0 KiB) TX bytes:2731 (2.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

# ping 10.42.0.1
PING 10.42.0.1 (10.42.0.1): 56 data bytes
64 bytes from 10.42.0.1: seq=0 ttl=64 time=0.933 ms
64 bytes from 10.42.0.1: seq=1 ttl=64 time=0.779 ms
```

Na host



```

user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
user@lab-44:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.9.138 netmask 255.255.255.0 broadcast 192.168.9.255
        inet6 fe80::2e80:35d%eno1 prefixlen 64 scopeid 0x20<link>
            ether ec:b1:d7:35:af:69 txqueuelen 1000 (Ethernet)
            RX packets 450438 bytes 653671621 (653.6 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 102326 bytes 8095495 (8.0 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
            device interrupt 20 memory 0xf7c00000-f7c20000

enx00e04c36003e: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.42.0.1 netmask 255.255.255.0 broadcast 10.42.0.255
        inet6 fe80::fc42:24a:ac6f:77b1 prefixlen 64 scopeid 0x20<link>
            ether 00:e0:4c:36:00:3e txqueuelen 1000 (Ethernet)
            RX packets 7 bytes 1831 (1.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 72 bytes 10496 (10.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 1757 bytes 166051 (166.0 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1757 bytes 166051 (166.0 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

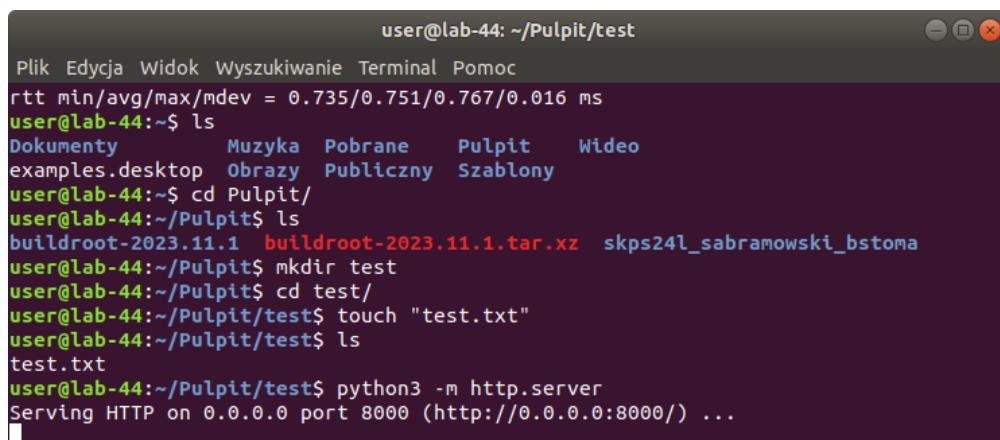
user@lab-44:~$ ping 10.42.0.63
PING 10.42.0.63 (10.42.0.63) 56(84) bytes of data.
64 bytes from 10.42.0.63: icmp_seq=1 ttl=64 time=0.767 ms
64 bytes from 10.42.0.63: icmp_seq=2 ttl=64 time=0.735 ms
^C
--- 10.42.0.63 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.735/0.751/0.767/0.016 ms
user@lab-44:~$ 

```

## Kopiowanie plików na RPi

Korzystaliśmy do przesyłania plików z metody z serwerem HTTP, na którym mieliśmy zawartość katalogu o nazwie 'test'

Tak utworzyliśmy serwer:

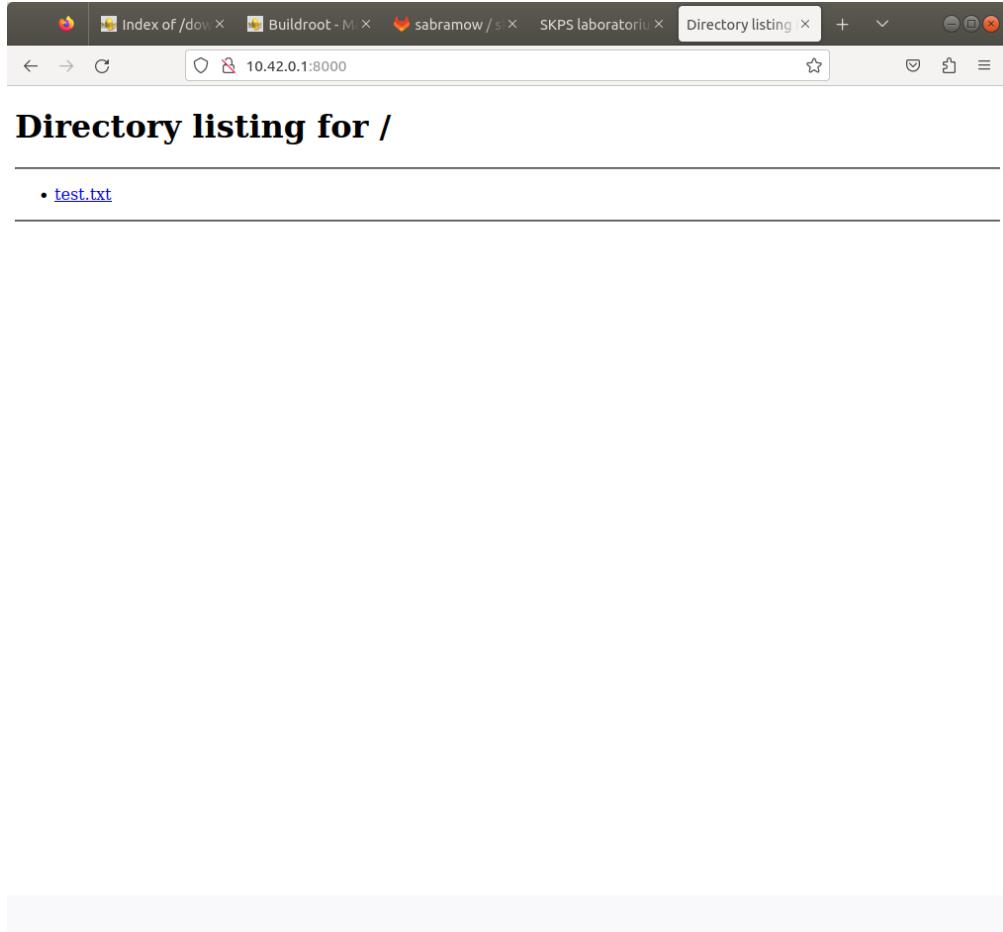


```

user@lab-44: ~/Pulpit/test
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
rtt min/avg/max/mdev = 0.735/0.751/0.767/0.016 ms
user@lab-44:~$ ls
Dokumenty Muzyka Pobrane Pulpit Wideo
examples.desktop Obrazy Publiczny Szablony
user@lab-44:~$ cd Pulpit/
user@lab-44:~/Pulpit$ ls
buildroot-2023.11.1 buildroot-2023.11.1.tar.xz skps24l_sabramowski_bstoma
user@lab-44:~/Pulpit$ mkdir test
user@lab-44:~/Pulpit$ cd test/
user@lab-44:~/Pulpit/test$ touch "test.txt"
user@lab-44:~/Pulpit/test$ ls
test.txt
user@lab-44:~/Pulpit/test$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...

```

Sprawdziliśmy jego działanie przez pojedynczy plik 'test.txt', który znajdował się w katalogu 'test'



## Kompilacja obrazu Linuxa w Buildroot

Korzystaliśmy z wersji Buildroot 2023.11.1, pobraliśmy ją przez 'wget  
<https://buildroot.org/downloads/buildroot-2023.11.1.tar.xz>'

### Obraz dla RPi z initramfs

Najpierw wykonaliśmy komendy:

```
make raspberrypi4_64_defconfig
make menuconfig
```

Ustawiliśmy Toolchain na External zgodnie z instrukcją oraz w ustawieniach 'Filesystem images' dokonaliśmy następujących zmian

```
Filesystem images
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
selects a feature, while <N> excludes a feature. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] feature is selected

[ ] axfs root filesystem
[ ] btrfs root filesystem
[ ] cloop root filesystem for the target device
-* [ ] cpio the root filesystem (for use as an initial RAM fileste
    cpio type (cpio the whole root filesystem) --->
    Compression method (gzip) --->
[ ] Create U-Boot image of the root filesystem
[ ] cramfs root filesystem
[ ] erofs root filesystem
[ ] ext2/3/4 root filesystem
[ ] f2fs root filesystem
[*] initial RAM filesystem linked into linux kernel
[ ] jffs2 root filesystem
[ ] oci image
[ ] romfs root filesystem
[ ] squashfs root filesystem
[ ] tar the root filesystem
[ ] ubi image containing an ubifs root filesystem
[ ] ubifs root filesystem
[ ] yaffs2 root filesystem

<Select> < Exit > < Help > < Save > < Load >
```

Następnie zbudowaliśmy obraz polecieniem

```
make
```

## Uruchomienie zbudowanego obrazu

Potem skopiowaliśmy potrzebne nam pliki do zbudowania obrazu do naszego folderu 'test', którego zawartość jest dostępna przez serwer HTTP

```
user@lab-44: ~/Pulpit/buildroot-2023.11.1/output/images
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
cmdline.txt config.txt fixup4.dat overlays start4.elf
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images/rpi-firmware$ cd ..
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ ls
bcm2711-rpi-4-b.dtb Image rootfs.cpio.gz
genimage.cfg rootfs.cpio rpi-firmware
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cd rpi-firmware/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images/rpi-firmware$ ls
cmdline.txt config.txt fixup4.dat overlays start4.elf
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images/rpi-firmware$ cd ..
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ ls
bcm2711-rpi-4-b.dtb Image rootfs.cpio.gz
genimage.cfg rootfs.cpio rpi-firmware
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cd rpi-firmware/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images/rpi-firmware$ cp cmdline.
txt /home/user/Pulpit/test
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images/rpi-firmware$ cd ..
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ ls
bcm2711-rpi-4-b.dtb Image rootfs.cpio.gz
genimage.cfg rootfs.cpio rpi-firmware
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp bcm2711-rpi-4-b.dtb /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp Image /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$
```

Sprawdziliśmy czy wszystko pliki są udostępnione

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Index of / | Buildroot - M | cw1 · main · S | SKPS laboratorium | Directory listing | + | -

← → 🔍 10.42.0.1:8000

## Directory listing for /

---

- [bcm2711-rpi-4-b.dtb](#)
- [cmdline.txt](#)
- [Image](#)
- [test.txt](#)

Następnie zamontowaliśmy partycję 1 karty SD w katalogu /mnt i sprawdziliśmy jego zawartość

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
# ls
bin      init     linuxrc  opt      run      tmp
dev      lib      media    proc    sbin    usr
etc      lib64   mnt     root    sys     var
# mount /dev/mmcblk0p1 /mnt
[ 2242.003761] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
# ls
bin      init     linuxrc  opt      run      tmp
dev      lib      media    proc    sbin    usr
etc      lib64   mnt     root    sys     var
# cd mnt
# ls
config.txt  fixup4.dat  overlays  rescue      start4.elf  user
# 
```

Pobraliśmy 3 wcześniej wrzucone na serwer pliki do RPi oraz zmieniliśmy nazwę obrazu z Image na kernel8.img

```
user@lab-44: ~

Plik Edycja Widok Wyszukiwanie Terminal Pomoc

# ls
bin      init     linuxrc  opt      run      tmp
dev      lib      media    proc    sbin    usr
etc      lib64   mnt     root    sys     var

# cd mnt
# ls
config.txt  fixup4.dat  overlays  rescue      start4.elf  user
# pwd
/mnt
# cd user/
# ls
# wget http://10.42.0.1:8000/Image
--1970-01-01 00:40:16-- http://10.42.0.1:8000/Image
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47477248 (45M) [application/octet-stream]
Saving to: 'Image'

Image          100%[=====] 45.28M 11.2MB/s  in 4.0s

1970-01-01 00:40:20 (11.2 MB/s) - 'Image' saved [47477248/47477248]

# wget http://10.42.0.1:8000/cmdline.txt
--1970-01-01 00:40:30-- http://10.42.0.1:8000/cmdline.txt
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 65 [text/plain]
Saving to: 'cmdline.txt'

cmdline.txt      100%[=====] 65  ---KB/s  in 0s

1970-01-01 00:40:30 (1.80 MB/s) - 'cmdline.txt' saved [65/65]

# wget http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
--1970-01-01 00:40:42-- http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51543 (50K) [application/octet-stream]
Saving to: 'bcm2711-rpi-4-b.dtb'

bcm2711-rpi-4-b.dtb 100%[=====] 50.33K  ---KB/s  in 0.004s

1970-01-01 00:40:42 (13.2 MB/s) - 'bcm2711-rpi-4-b.dtb' saved [51543/51543]

# ls
Image          bcm2711-rpi-4-b.dtb  cmdline.txt
# mv ./Image ./kernel8.img
# ls
bcm2711-rpi-4-b.dtb  cmdline.txt      kernel8.img
#
```

Następnie uruchomiliśmy zbudowany system (trzymając przycisk SW4 zgodnie z instrukcją) oraz zapisałyśmy w nim jeden plik testowy

```

user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
udhcpc: broadcasting discover
[    8.286208] bcmgenet fd580000.ethernet eth0: Link is Up - 100Mbps/Full - flow
control rx/tx
udhcpc: no lease, forking to background
OK

Welcome to Buildroot
buildroot login: root
# ls
# touch "test.txt"
# ls[  35.805985] cam-dummy-reg: disabling
[  35.809606] cam1-reg: disabling

test.txt
# pwd
/root
# cd ..
# ls
bin      init      linuxrc  opt       run       tmp
dev      lib       media    proc     sbin     usr
etc      lib64     mnt     root     sys      var
# echo "test" > test.txt
# cat /.test.txt
cat: can't open '/.test.txt': No such file or directory
# cat test.txt
test
# 

```

Po restarcie systemu, pliku już nie było, co wskazywało na to, że system prawidłowo działał

```

user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
[    5.126064] mmc0: new ultra high speed DDR50 SDHC card at address 0001
Saving 2048 bits of non-creditable seed f[    5.134261] mmcblk0: mmc0:0001 SD32G
28.9 GiB
or next boot
Starting syslogd: [    5.142725] mmcblk0: p1 p2
OK
Starting klogd: OK
Running sysctl: OK
Starting network: [    5.243121] bcmgenet fd580000.ethernet: configuring instance for external RGMII (RX delay)
[    5.252071] bcmgenet fd580000.ethernet eth0: Link is Down
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
[    8.318219] bcmgenet fd580000.ethernet eth0: Link is Up - 100Mbps/Full - flow
control rx/tx
udhcpc: no lease, forking to background
OK

Welcome to Buildroot
buildroot login: root
# ls
# cd ..
# ls
bin      init      linuxrc  opt       run       tmp
dev      lib       media    proc     sbin     usr
etc      lib64     mnt     root     sys      var
# 

```

## Obraz dla RPi bez initramfs

Wykonaliśmy komende w celu czyszenia po poprzednim obrazie według instrukcji i odpaliliśmy menu do konfiguracji

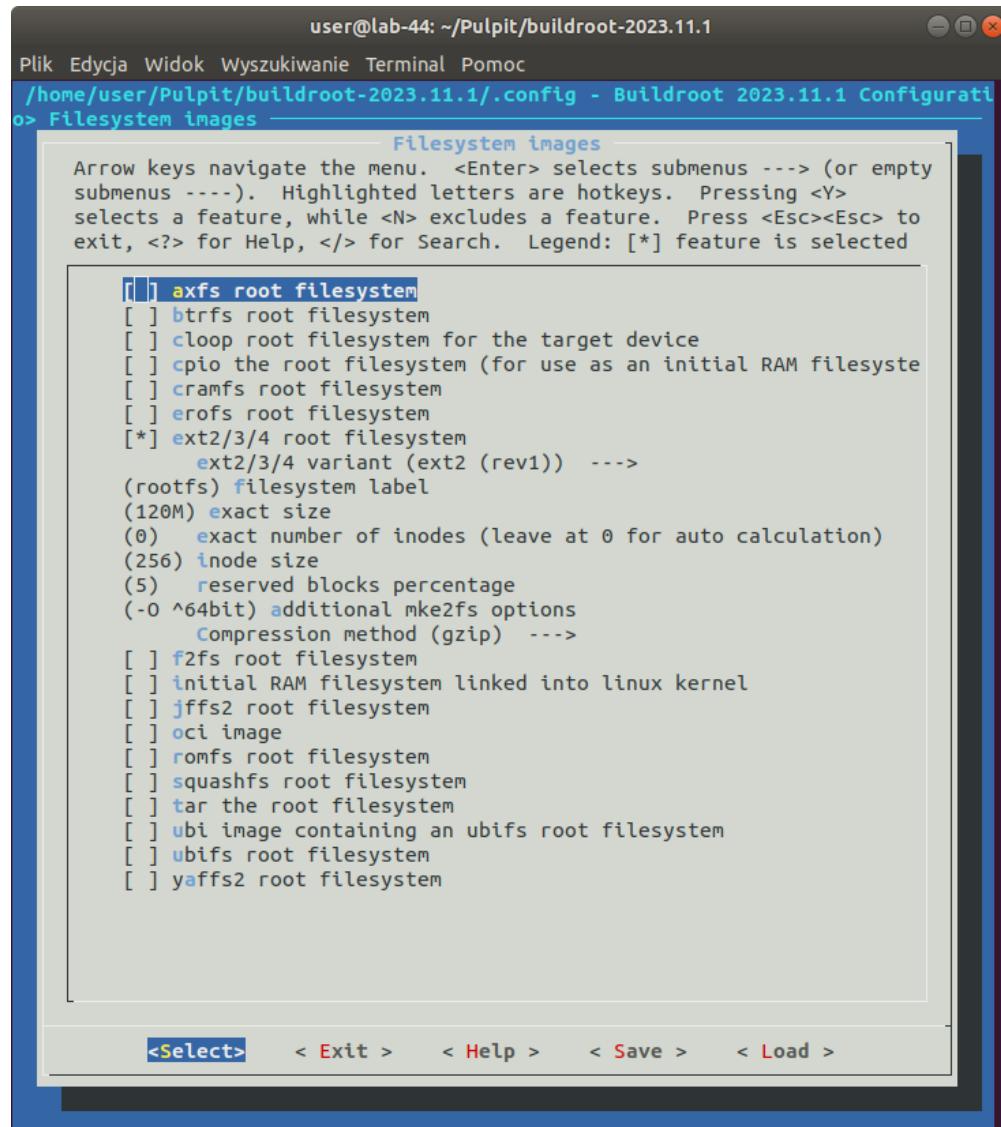
```

make linux-dirclean
make menuconfig

```

Zostawiliśmy Toolchain ustawiony na External, wyłączyliśmy initramfs, włączyliśmy wsparcie dla ext2/3/4 (ext2 wybraliśmy) oraz pozostawiliśmy ustawienie kompresji na gzip

Dodatkowo dwukrotnie zwiększyliśmy wielkość systemu plików



Wykonaliśmy komplikację systemu przez

```
make clean all
```

Ponownie przerzuciliśmy wymagane pliki do udostępnionego folderu (tym razem także plik z systemem plików)

```
user@lab-44: ~/Pulpit/buildroot-2023.11.1/output/images
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
Copying files into the device: __populate_fs: Could not allocate block in ext2 filesystem while writing file "nilfs2.ko"
mkfs.ext2: Could not allocate block in ext2 filesystem while populating file system
*** Maybe you need to increase the filesystem size (BR2_TARGET_ROOTFS_EXT2_SIZE)
fs/ext2/ext2.mk:47: recipe for target '/home/user/Pulpit/buildroot-2023.11.1/output/images/rootfs.ext2' failed
make[1]: *** [/home/user/Pulpit/buildroot-2023.11.1/output/images/rootfs.ext2] Error 1
Makefile:82: recipe for target '_all' failed
make: *** [_all] Error 2
user@lab-44:~/Pulpit/buildroot-2023.11.1$ cd output/images/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ ls
bcm2711-rpi-4-b.dtb  Image  rootfs.cpio.gz  rpi-firmware
genimage.cfg          rootfs.cpio  rootfs.ext2
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp Image /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp rootfs.ext2 /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp bcm2711-rpi-4-b.dtb /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$ cp rpi-firmware/cmdline.txt /home/user/Pulpit/test/
user@lab-44:~/Pulpit/buildroot-2023.11.1/output/images$
```

Sprawdziliśmy zawartość udostępnionego folderu



The screenshot shows a web browser window with the address bar set to 10.42.0.1:8000. The page title is "Directory listing for /". The content of the page is a list of files and directories:

- [bcm2711-rpi-4-b.dtb](#)
- [cmdline.txt](#)
- [Image](#)
- [ramdisk\\_bcm2711-rpi-4-b.dtb](#)
- [ramdisk\\_cmdline.txt](#)
- [ramdisk\\_Image](#)
- [rootfs.ext2](#)

Wrzuciliśmy plik 'rootfs.ext2' na partycję drugą karty SD

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
# [ 35.805054] cam-dummy-reg: disabling
[ 35.808675] cam1-reg: disabling

# wget http://10.42.0.1:8000/rootfs.ext2
--1970-01-01 00:02:14-- http://10.42.0.1:8000/rootfs.ext2
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 125829120 (120M) [application/octet-stream]
Saving to: 'rootfs.ext2'

rootfs.ext2      100%[=====] 120.00M 11.2MB/s    in 11s

1970-01-01 00:02:25 (11.2 MB/s) - 'rootfs.ext2' saved [125829120/125829120]

# ls
bin      lib      mnt      rootfs.ext2  tmp
dev      lib64    opt      run       usr
etc      linuxrc  proc     sbin      var
init    media    root     sys
# dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096
30720+0 records in
30720+0 records out
# 
```

Zamontowaliśmy pierwszą partycję karty SD jak poprzednio i dodaliśmy do niej wymagane pliki (pamiętając o tym żeby zmienić nazwę obrazu z Image na kernel8.img)

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
# ls
adapter_test.sh
# cd .
# ls
adapter_test.sh
# cd ..
# ls
bin      init    linuxrc  opt      run      tmp
dev      lib     media    proc     sbin     usr
etc      lib64   mnt     root     sys      var
# cd mnt
# ls
# ls
# mount /dev/mmcblk0p1 /mnt
[ 427.470840] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
# ls
# pwd
/mnt
# cd ..
# ls
bin      init    linuxrc  opt      run      tmp
dev      lib     media    proc     sbin     usr
etc      lib64   mnt     root     sys      var
# cd mnt/
# ls
config.txt  fixup4.dat  overlays  rescue      start4.elf  user
# cd user/
# ls
bcm2711-rpi-4-b.dtb  cmdline.txt          kernel8.img
# rm ./*
# ls
# pwd
/mnt/user
# wget http://10.42.0.1:8000/Image
--1970-01-01 00:08:38--  http://10.42.0.1:8000/Image
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21656064 (21M) [application/octet-stream]
Saving to: 'Image'

Image           100%[=====] 20.65M 11.1MB/s  in 1.9s

1970-01-01 00:08:40 (11.1 MB/s) - 'Image' saved [21656064/21656064]

# wget http://10.42.0.1:8000/cmdline.txt
--1970-01-01 00:08:51--  http://10.42.0.1:8000/cmdline.txt
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 65 [text/plain]
Saving to: 'cmdline.txt'

cmdline.txt       100%[=====] 65 ---KB/s  in 0s
1970-01-01 00:08:51 (914 KB/s) - 'cmdline.txt' saved [65/65]

# wget http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
--1970-01-01 00:09:16--  http://10.42.0.1:8000/bcm2711-rpi-4-b.dtb
Connecting to 10.42.0.1:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51543 (50K) [application/octet-stream]
Saving to: 'bcm2711-rpi-4-b.dtb'

bcm2711-rpi-4-b.dtb 100%[=====] 50.33K ---KB/s  in 0.004s
1970-01-01 00:09:16 (13.3 MB/s) - 'bcm2711-rpi-4-b.dtb' saved [51543/51543]

# mv Image kernel8.img
# ls
bcm2711-rpi-4-b.dtb  cmdline.txt          kernel8.img
#
```

Przetestowaliśmy działanie systemu z normalnym systemem plików poprzez zapisanie w nim pliku testowego

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
Starting klogd: OK
Running sysctl: OK
Starting network: [    3.381359] bcmgenet fd580000.ethernet: configuring instance for external RMII (RX delay)
[    3.390239] bcmgenet fd580000.ethernet eth0: Link is Down
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
[    6.462308] bcmgenet fd580000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
udhcpc: no lease, forking to background
OK

Welcome to Buildroot
buildroot login: root
# ls
# cd ..
# echo "test" > test.txt
# ls
bin      lib64      mnt      run      tmp
dev      linuxrc      opt      sbin      usr
etc      lost+found  proc      sys      var
lib      media      root      test.txt
# 
```

Po restarcie systemu, plik oraz jego zawartość były dostępne, więc system bez initramfs działał prawidłowo

```
user@lab-44: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
Running sysctl: OK
Starting network: [    3.373819] bcmgenet fd580000.ethernet: configuring instance for external RMII (RX delay)
[    3.382592] bcmgenet fd580000.ethernet eth0: Link is Down
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
[    6.462296] bcmgenet fd580000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
udhcpc: no lease, forking to background
OK

Welcome to Buildroot
buildroot login: root
# ls
# cd ..
# ls
bin      lib64      mnt      run      tmp
dev      linuxrc      opt      sbin      usr
etc      lost+found  proc      sys      var
lib      media      root      test.txt
# cat test.txt
test
# 
```

## Porównanie wielkości obrazów

Rozmiar obrazu systemu z initramfs

```
user@lab-44: ~/Pulpit/test
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
user@lab-44:~$ cd Pu
Publiczny/ Pulpit/
user@lab-44:~$ cd Pu
Publiczny/ Pulpit/
user@lab-44:~$ cd Pulpit/test/
user@lab-44:~/Pulpit/test$ ls
bcm2711-rpi-4-b.dtb cmdline.txt Image test.txt
user@lab-44:~/Pulpit/test$ ls -a
. .. bcm2711-rpi-4-b.dtb cmdline.txt Image test.txt
user@lab-44:~/Pulpit/test$ man ls
user@lab-44:~/Pulpit/test$ ls -l
razem 46280
-rwxr-xr-x 1 user user 51543 mar 6 13:27 bcm2711-rpi-4-b.dtb
-rw-r--r-- 1 user user 65 mar 6 13:26 cmdline.txt
-rw-r--r-- 1 user user 47477248 mar 6 13:27 Image
-rw-rw-r-- 1 user user 0 mar 6 13:02 test.txt
user@lab-44:~/Pulpit/test$ du -h Image
46M Image
user@lab-44:~/Pulpit/test$ 
```

Rozmiar obrazu systemu ze zwykłym systemem plików

```
user@lab-44: ~/Pulpit/test
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
user@lab-44:~/Pulpit/test$ ls -l
razem 187068
-rwxr-xr-x 1 user user 51543 mar 6 15:07 bcm2711-rpi-4-b.dtb
-rw-r--r-- 1 user user 65 mar 6 15:07 cmdline.txt
-rwxr-xr-x 1 user user 51543 mar 6 14:15 er_bcm2711-rpi-4-b.dtb
-rw-r--r-- 1 user user 65 mar 6 14:15 er_cmdline.txt
-rw-r--r-- 1 user user 112355 mar 6 14:17 er_config_zwykly
-rw-r--r-- 1 user user 21656064 mar 6 14:14 er_Image
-rw-r--r-- 1 user user 21656064 mar 6 15:06 Image
-rwxr-xr-x 1 user user 51543 mar 6 13:27 randysk_bcm2711-rpi-4-b.dtb
-rw-r--r-- 1 user user 65 mar 6 13:26 randysk_cmdline.txt
-rw-r--r-- 1 user user 47477248 mar 6 13:27 randysk_Image
-rw-r--r-- 1 user user 125829120 mar 6 15:06 rootfs.ext2
user@lab-44:~/Pulpit/test$ du -h Image
21M Image
user@lab-44:~/Pulpit/test$ 
```

Czyli rozmiar obrazu systemu z initramfs był ponad dwa razy większy. Wynika to z tego, że startowy system plików jest wrzucony do środka obrazu systemu z initramfs. W systemie ze zwykłym systemie plików, system plików zajmuje miejsce na karcie SD.