

SKPS - Laboratorium 1

Zespół korzysta z karty SD - 105e

1. Przygotowanie stanowiska

1. Płytką została podłączona zgodnie ze schematem z instrukcji
2. Prowadzący zweryfikował poprawność podłączenia
3. Po włączeniu zasilania w terminalu pokazały się logi z bootowania

```
user@lab-22:~$ tto /dev/ttyUSB0
[tto 16:34:56] tto v1.29
[tto 16:34:56] Press ctrl-t q to quit
[tto 16:34:56] Connected

PM_RSTS: 0x00001000
RPI: BOOTLOADER release VERSION:c2f8c388 DATE: Apr 29 2021 TIME: 17:11:29 BOOTMODE: 0x00000006 part: 0 BUILD_TIMESTAMP=1619712685 0x913e7900 0x00c03114 0x000b057b
PM_RSTS: 0x00001000
part 00000000 reset_info 00000000
USD voltage 3.3V
Initialising SDRAM 'Samsung' 16Gb x2 total-size: 32 Gbit 3200
PCI reset
PCI reset
VLI: HUB2: 0xffff0000 0x24e6 MCU: 0xffff20000 0x15218
XHCI-STOP
```

4. Po zalogowaniu ręcznie uruchomiliśmy DHCP

```
Welcome to Buildroot rescue OS
rescue login: root
# udhcpd
udhcpd: started, v1.33.1
udhcpd: sending discover
udhcpd: sending select for 10.42.0.155
udhcpd: lease of 10.42.0.155 obtained, lease time 3600
deleting routers
adding dns 10.42.0.1
```

5. Istnieje możliwość pingowania

```
user@lab-22:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.9.117 netmask 255.255.255.0 broadcast 192.168.9.255
    inet6 fe80::637b:b9f:9abd:4213 prefixlen 64 scopeid 0x20<link>
    ether ec:b1:d7:31:3b:3d txqueuelen 1000 (Ethernet)
    RX packets 682491 bytes 1006344717 (1.0 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 162333 bytes 20373121 (20.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7d00000-f7d20000

enx00e04c3601fc: 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::9511:a53e:c971:82e9 prefixlen 64 scopeid 0x20<link>
    ether 00:e0:4c:36:01:fc txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 1831 (1.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 109 bytes 13819 (13.8 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 6378 bytes 901352 (901.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6378 bytes 901352 (901.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user@lab-22:~$ ping 10.42.0.1
PING 10.42.0.1 (10.42.0.1) 56(84) bytes of data.
64 bytes from 10.42.0.1: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 10.42.0.1: icmp_seq=2 ttl=64 time=0.020 ms
64 bytes from 10.42.0.1: icmp_seq=3 ttl=64 time=0.023 ms
^C
--- 10.42.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2048ms
rtt min/avg/max/mdev = 0.020/0.022/0.025/0.005 ms
user@lab-22:~$
```

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr E4:5F:01:2B:50:80
          inet addr:10.42.0.155  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::e65f:1ff:fe2b:5080/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2716 (2.6 KiB)  TX bytes:2981 (2.9 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 192.168.9.117
PING 192.168.9.117 (192.168.9.117): 56 data bytes
64 bytes from 192.168.9.117: seq=0 ttl=64 time=0.918 ms
64 bytes from 192.168.9.117: seq=1 ttl=64 time=0.784 ms
64 bytes from 192.168.9.117: seq=2 ttl=64 time=0.751 ms
^C
--- 192.168.9.117 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.751/0.817/0.918 ms
#
```

2. Obraz systemu z initramfs

1. Ze strony buildroot.org pobraliśmy i wypakowaliśmy program buildroot w wersji 2021.08
2. Następnie w katalogu z rozpakowaną paczką uruchomiliśmy następujące komendy:
make raspberrypi4_64_defconfig
make menuconfig
3. W menu zaznaczyliśmy odpowiednio:
 - Toolchain type: External toolchain
 - Initial RAM filesystem linked...
 - Cpio the root filesystem -> compression method -> gzip
 A odznaczona została opcja:
 - ext 2/3/4 filesystem
4. Po wykonaniu wyżej opisanych działań i zapisaniu konfiguracji w katalogu z rozpakowaną paczką uruchomiliśmy kompilację przy pomocy komendy:
make
5. Udało nam się skompilować jądro, zakończyła się jednak błędem niewykrywania pliku rootfs.ext.4 - co jest w tym przypadku normalne, bo wyłączyliśmy korzystanie z systemu plików ext4.

```
*** Executing post-image script board/raspberrypi4-64/post-image.sh
[INFO] device is not booting (0 configfs) (lines 1-1000) (all console)
board/raspberrypi4-64/genimage-raspberrypi4-64.cfg:31: no sub-section title/index for 'config'
[INFO] cmd: "mkdir -p /home/user/buildroot-2021.08/output/build/genimage.tmp" (stderr):
[INFO] cmd: "cp -r /home/user/buildroot-2021.08/output/build/genimage.tmp/*" (stderr):
[INFO] cmd: "mkdir -p /home/user/buildroot-2021.08/output/build/genimage.tmp" (stderr):
[INFO] cmd: "cp -r /tmp/tmp.y4z8p8s8 /home/user/buildroot-2021.08/output/build/genimage.tmp/root" (stderr):
[INFO] cmd: "find /home/user/buildroot-2021.08/output/build/genimage.tmp/root -depth -type d -printf '%P\n' | xargs -0 -i {} touch -r '/tmp/tmp.y4z8p8s8/{}' /home/user/buildroot-2021.08/output/build/g
enimage.tmp/root/{}" (stderr):
[ERROR] file(rootfs.ext4): stat(/home/user/buildroot-2021.08/output/images/rootfs.ext4) failed: No such file or directory
[ERROR] hdimage(sdcard.img): could not setup rootfs.ext4
makefile:22: recipe for target 'target-post-image' failed
make[1]: *** [target-post-image] Error 1
makefile:80: recipe for target '_all' failed
make: *** [_all] Error 2
user@lab-22:~/buildroot-2021.08
```

6. Partycję 1 (boot) karty SD zamontowaliśmy w katalogu /mnt za pomocą polecenia:
mount /dev/mmcblk0p1 /mnt
7. Przed restartem RPi skopiowaliśmy pliki Image (jako kernel8.img) , cmdline.txt i bcm2711-rpi-4-b.dtb do katalogu /mnt/user wykorzystując do tego serwer http.

```
# mount /dev/mmcblk0p1 /mnt
[ 4387.996599] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
# cd /mnt
# ls
config.txt  fixup4.dat  overlays  rescue  start4.elf  user
# wget http://192.168.9.117:8000/output/images/Image
--1970-01-01 01:15:00-- http://192.168.9.117:8000/output/images/Image
Connecting to 192.168.9.117:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47213056 (45M) [application/octet-stream]
Saving to: 'Image'

Image          100%[=====] 45.03M  11.2MB/s   in 4.0s

1970-01-01 01:15:04 (11.2 MB/s) - 'Image' saved [47213056/47213056]

# wget http://192.168.9.117:8000/package/rpi-firmware/cmdline.txt
--1970-01-01 01:18:20-- http://192.168.9.117:8000/package/rpi-firmware/cmdline.txt
Connecting to 192.168.9.117:8000... 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 01:18:20 (1.91 MB/s) - 'cmdline.txt' saved [65/65]

# wget http://192.168.9.117:8000/output/images/bcm2711-rpi-4-b.dtb
--1970-01-01 01:20:36-- http://192.168.9.117:8000/output/images/bcm2711-rpi-4-b.dtb
Connecting to 192.168.9.117:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49749 (49K) [application/octet-stream]
Saving to: 'bcm2711-rpi-4-b.dtb'

bcm2711-rpi-4-b.dtb 100%[=====] 48.58K  --.-KB/s   in 0.003s

1970-01-01 01:20:36 (13.6 MB/s) - 'bcm2711-rpi-4-b.dtb' saved [49749/49749]

# mv Image bcm2711-rpi-4-b.dtb cmdline.txt ./user
# ls
config.txt  fixup4.dat  overlays  rescue  start4.elf  user
# cd user
# ls
Image          bcm2711-rpi-4-b.dtb  cmdline.txt
# mv Image kernel8.img
# ls
bcm2711-rpi-4-b.dtb  cmdline.txt          kernel8.img
#
```

8. Po restarcie RPI (poleceniem **reboot**) system uruchomił się poprawnie

```
Welcome to Buildroot
buildroot login: root
# touch grass.txt
# ls
grass.txt
# pwd
/root
```

Utworzony wyżej plik po zrestartowaniu systemu został usunięty - system działa poprawnie.

```
Welcome to Buildroot
buildroot login: root
# ls
# pwd
/root
#
```

3. Obraz systemu bez initramfs

1. W katalogu z rozpakowanym buildrootem zaczynamy od usunięcia poprzedniego obrazu:

```
make linux-dirclean
```


2. Uruchomiliśmy ponownie komendy:

```
make raspberrypi4_64_defconfig
```

```
make menuconfig
```

3. W menu zaznaczyliśmy:

- Toolchain type: External toolchain
- Wsparcie dla ext2 (rev1)
- Kompresję plików w formacie gzip

Wyłączone zostało wsparcie dla initramfs.

Zwiększyliśmy rozmiar partycji boot, w budowanym obrazie karty SD na 256M

4. Po wykonaniu wyżej opisanych działań i zapisaniu konfiguracji w katalogu z rozpakowaną paczką uruchomiliśmy kompilację przy pomocy komendy:

make

5. Kompilacja została zakończona sukcesem

```

root@raspberrypi-06:/gentage-raspberrypi-04-ef313# make sub action index for 'config'
make[2]: /home/user/builderoot-2021.08/output/build/gentage-mpm/*~ (stderr)
make[2]: *** No rule to make target '/home/user/builderoot-2021.08/output/build/gentage-mpm/*~'.  Stop.
make[1]: *** [/cp -s /home/user/builderoot-2021.08/output/build/gentage-mpm/root~] (stderr)
make[1]: *** No rule to make target '/home/user/builderoot-2021.08/output/build/gentage-mpm/root~'.  Stop.
make[1]: *** No rule to make target '/home/user/builderoot-2021.08/output/images/rootfs.ext4' failed: No such file or directory
make[1]: *** Could not setup rootsfs.ext4
makefile:121: recipe for target 'target-post-image' failed
make: *** [target-post-image] Error 1
makefile:114: recipe for target 'all' failed
make: *** [all] Error 1
makefile:22: /builderoot-2021.085

```

- 6.** Partycję 1 (boot) karty SD zamontowaliśmy w katalogu /mnt za pomocą polecenia:

```
mount /dev/mmcblk0p1 /mnt
```

7. Przed restartem RPi skopiowaliśmy pliki Image (jako kernel8.img), cmdline.txt i bcm2711-rpi-4-b.dtb do katalogu /mnt/user wykorzystując do tego serwer http.

```

# wget http://192.168.9.117:8000/output/images/Image
--1970-01-01 00:02:01-- http://192.168.9.117:8000/output/images/Image
Connecting to 192.168.9.117:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21719552 (21M) [application/octet-stream]
Saving to: 'Image'

Image                               100%[=====] 20.71M  11.1MB/s   in 1.9s

1970-01-01 00:02:02 (11.1 MB/s) - 'Image' saved [21719552/21719552]

# wget http://192.168.9.117:8000/package/rpi-firmware/cmdline.txt
--1970-01-01 00:02:20-- http://192.168.9.117:8000/package/rpi-firmware/cmdline.txt
Connecting to 192.168.9.117:8000... 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 00:02:20 (1.88 MB/s) - 'cmdline.txt' saved [65/65]

# wget http://192.168.9.117:8000/output/images/bcm2711-rpi-4-b.dtb
--1970-01-01 00:02:46-- http://192.168.9.117:8000/output/images/bcm2711-rpi-4-b.dtb
Connecting to 192.168.9.117:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49749 (49K) [application/octet-stream]
Saving to: 'bcm2711-rpi-4-b.dtb'

bcm2711-rpi-4-b.dtb 100%[=====] 48.58K  --.-KB/s   in 0.004s

1970-01-01 00:02:46 (13.3 MB/s) - 'bcm2711-rpi-4-b.dtb' saved [49749/49749]

# wget http://192.168.9.117:8000/output/images/rootfs.ext2
--1970-01-01 00:03:15-- http://192.168.9.117:8000/output/images/rootfs.ext2
Connecting to 192.168.9.117:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 62914560 (60M) [application/octet-stream]
Saving to: 'rootfs.ext2'

rootfs.ext2                     100%[=====] 60.00M  11.2MB/s   in 5.3s

1970-01-01 00:03:21 (11.2 MB/s) - 'rootfs.ext2' saved [62914560/62914560]

```

Dodatkowo na kartę SD został przesłany również plik *rootfs.ext2*, a następnie nagralismy system plików na partycje 2 (rootfs) przy pomocy komendy:

```
dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096
```

```
# mv Image kernel8.img
# pwd
/mnt/user
# dd if=rootfs.ext2 of=/dev/mmcblk0p2 bs=4096
15360+0 records in
15360+0 records out
```

8. Po restarcie RPi (poleceniem **reboot**) system uruchomił się poprawnie
Utworzony plik po zrestartowaniu systemu został zachowany - system działa poprawnie.

```
Welcome to Buildroot
buildroot login: root
# ls
grass.txt
# pwd
/root
#
```