

# Systemy komputerowe w sterowaniu i pomiarach (SKPS)

## Instrukcja do laboratorium 1

dr hab. inż. Wojciech Zabołotny, mgr inż. Dawid Seredyński

### Informacje wstępne

[Cel laboratorium](#)

[Zadanie domowe](#)

[Zakres wejściówka](#)

[Zadania](#)

### Sprzęt

[Karta SD](#)

[Zestaw laboratoryjny](#)

[Repozytorium i kody źródłowe](#)

[Oznaczenia użyte w tym dokumencie](#)

### Przygotowanie stanowiska

[Pierwsze uruchomienie RPi](#)

[Kopiowanie plików na RPi](#)

[Logowanie za pomocą ssh systemie ratunkowym na RPi](#)

### Kompilacja obrazu Linuxa w Buildroot

[Obraz dla Raspberry Pi 4B z initramfs](#)

[Uruchomienie zbudowanego obrazu](#)

[Obraz dla Raspberry Pi 4B bez initramfs](#)

[Zadanie domowe: Obraz na 64-bitową maszynę wirtualną "Virt" i uruchomienie w QEMU](#)

## Informacje wstępne

### Cel laboratorium

Celem laboratorium jest zapoznanie się z Buildroot oraz uruchomienie go w QEMU i na RPi. Laboratorium 1 jest za 8 punktów, w tym:

- 0 - zadanie domowe - bez punktacji, ale zrobienie zadania jest bardzo pomocne w pracy na laboratorium
- 2 - wejściówka
- 6 - praca na zajęciach

## Zadanie domowe

Zadanie domowe należy wykonać przed zajęciami

1. zbudowanie za pomocą Buildroot obrazu Linuxa dla maszyny wirtualnej i uruchomienie na qemu: [Zadanie domowe: Obraz na 64-bitową maszynę wirtualną "Virt" i uruchomienie w QEMU](#)

## Zakres wejściówka

Na wejściówce można się spodziewać pytań z wykładu 1 oraz z niniejszej instrukcji, m.in.

1. qemu
2. Buildroot
3. transfer plików między dwoma komputerami w jednej sieci lokalnej

## Zadania

Na laboratorium 1 wykonywane są następujące zadania:

1. Złożenie stanowiska laboratoryjnego: zestaw z Raspberry Pi 4B (RPi),
2. Pierwsze uruchomienie RPi, sprawdzenie połączenia sieciowego, wykonanie próbnych transferów plików,
3. Zbudowanie za pomocą Buildroot obrazu Linuxa dla RPi, z init RAM fs,
4. Zbudowanie za pomocą Buildroot obrazu Linuxa dla RPi, z systemem plików na trwałym nośniku.

**UWAGA:** w czasie trwania zajęć należy przygotować raport - co i jak zostało zrobione, jakie były efekty.

## Sprzęt

### Karta SD

Na karcie SD, z której korzysta RPi, są 2 partycje:

1. **boot** - zawiera:
  - a. katalog overlays - overlays dla DT, czyli nakładki na drzewo urządzeń
  - b. katalog rescue - obraz systemu ratunkowego (Buildroot z initramfs), linia poleceń i drzewo urządzeń
  - c. katalog user - miejsce na obraz systemu, który zostanie wygenerowany w ramach zajęć
  - d. config.txt - plik z konfiguracją dla bootloadera
  - e. start4.elf - firmware GPU
  - f. fixup4.dat - plik związany z firmware
2. **rootfs** - system plików dla systemu wygenerowanego w trakcie zajęć

Studenci mogą zmieniać tylko zawartość katalogu /user na partycji nr 1 (boot) oraz partycję 2 (rootfs). Zmienianie pozostałych plików i katalogów może spowodować awarię i konieczność ponownego wgrywania całego obrazu karty SD.

## Zestaw laboratoryjny

Preferujemy restartowanie RPi za pomocą polecenia **reboot** (w Linuxie). Jeśli nie ma takiej możliwości (np. system się zawiesił), to restartujemy poprzez wyłączenie zasilania, odczekanie kilku sekund i włączenie zasilania na **listwie zasilającej RPi. Nie odłączamy kabla zasilającego, gdyż spowoduje to szybkie zużycie gniazda!**

Wszelkie zmiany w połączeniach wykonujemy przy wyłączonym zasilaniu RPi. Przed włączeniem RPi w nowej konfiguracji, należy za każdym razem powiadomić prowadzącego i zaczekać, aż sprawdzi poprawność połączeń.

Przed zakończeniem zajęć należy zwrócić zestawy w takim samym stanie, w jakim zostały wydane: należy zwinać i spiąć kable, spakować elementy do pudełek i opakowań.

## Repozytorium i kody źródłowe

Każdy zespół jest zobowiązany do utworzenia repozytorium na wydziałowym serwerze gitlab:

<https://gitlab-stud.elka.pw.edu.pl>

Nazwa repozytorium:

*skps23l\_inazwisko1\_inazwisko2*

*inazwisko* - pierwsza litera imienia i nazwisko, bez polskich liter, np. nazwa repozytorium dla zespołu Jana Kowalskiego i Anny Przykładowskiej to:

*skps23l\_jkowalski\_aprzykładowska*

W ww. repozytorium należy przechowywać:

- kody źródłowe (\*.c, \*.py, \*.h, Makefile, xml, itp.)
- pliki konfiguracyjne (.config)
- dokumentację: tylko w formacie markdown (\*.md) lub pdf

W repozytorium NIE przechowujemy:

- skompilowanych pakietów
- paczek ściągniętych z internetu (Buildroot, OpenWRT, SKD, itp.)

Struktura katalogów w repozytorium powinna być następująca::

- cw1
  - [zadanie domowe, kody i raporty dla ćwiczenia 1, pliki konfiguracyjne (.config)  
Buildroot]
- cw2
  - [zadanie domowe, kody i raporty dla ćwiczenia 2]
- cw3
  - [j.w.]
- cw4
  - [j.w.]
- projekt
  - [zadanie domowe, kody i raporty związane z projektem]

**UWAGA:** należy dodać osobę prowadzącą zajęcia do listy osób uprawnionych do przeglądania repozytorium.

**UWAGA 2:** należy wgrywać pliki do repozytorium w rozsądny sposób, np. nie należy wgrywać gigabajtów danych ze skompilowanym Buildroot, ale wystarczy wgrać plik z konfiguracją (.config) i opisać w raporcie, dla jakiej wersji Buildroot jest ta konfiguracja.

## Oznaczenia użyte w tym dokumencie

Polecenia, które są wpisywane w konsoli **Linuxa** na komputerze **host (PC)** oznaczone są jasnoszarym tłem, np.:

`ls -la /`

Polecenia, które są wpisywane w konsoli **Linuxa** na komputerze **RPi** oznaczone są malinowym tłem (raspberry, czyli malina), np.:

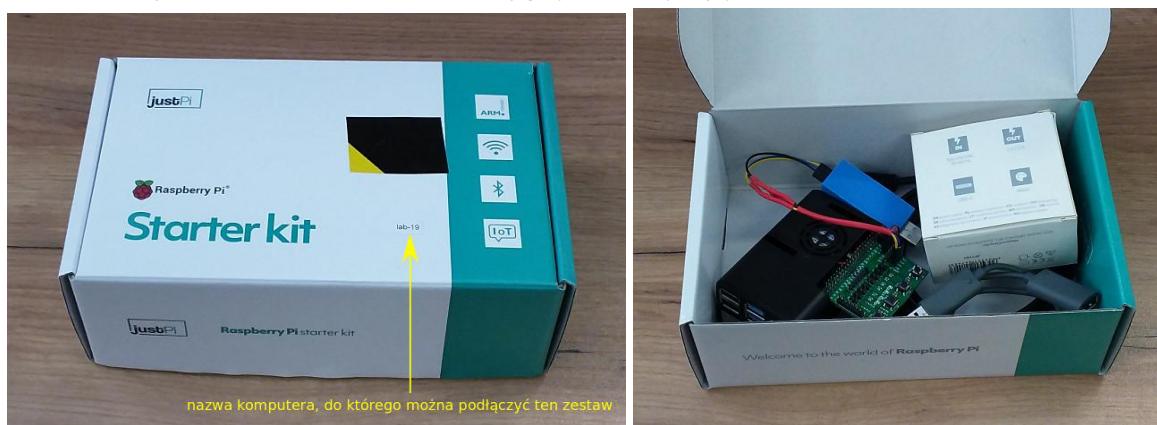
`ls -la /`

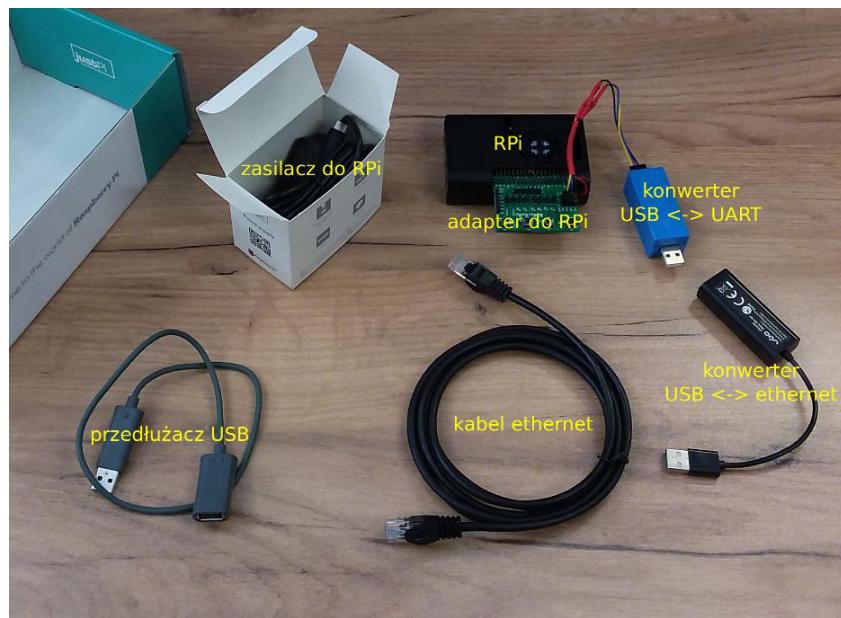
## Przygotowanie stanowiska

Studenci samodzielnie przygotowują stanowisko, podłączając urządzenia według instrukcji, ale **przed jego uruchomieniem (włączeniem zasilania) muszą skonsultować się z prowadzącym.**

**UWAGA:** konwertery USB/eth są związane z konkretnymi komputerami w sali P113 i P114. Proszę zwrócić uwagę na oznaczenia naklejone na ww. urządzenia i na komputery.

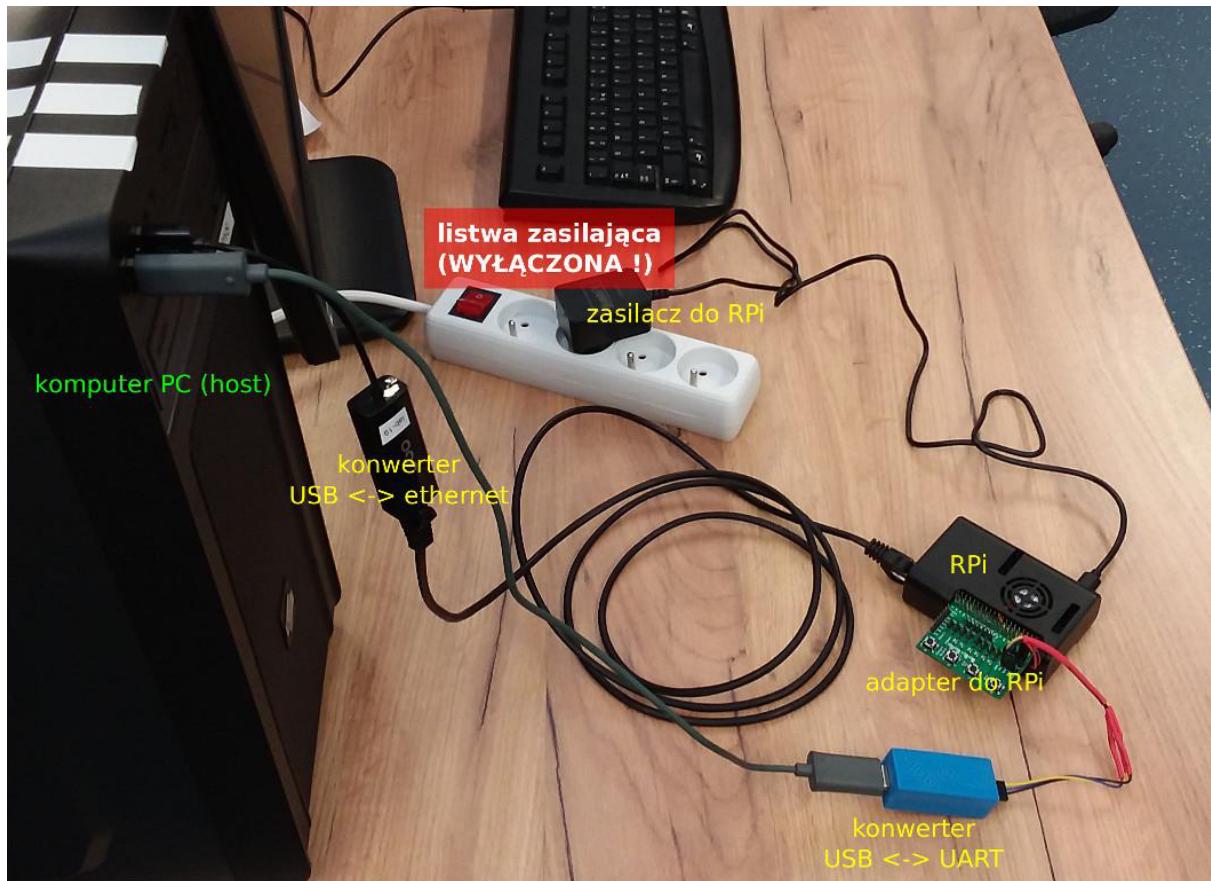
Podstawowy zestaw do laboratorium wygląda następująco:



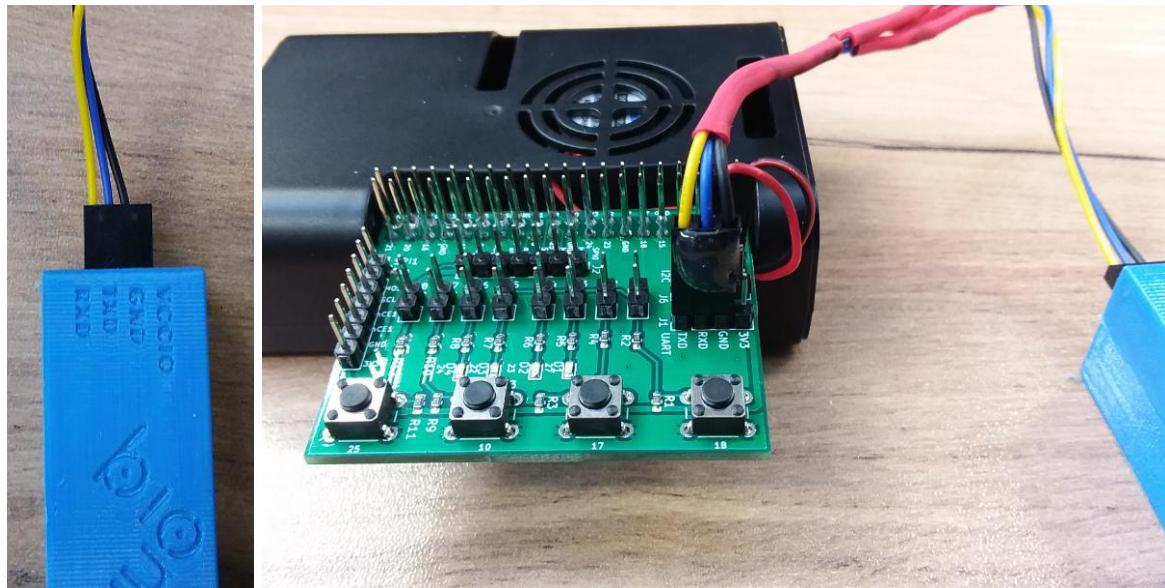


**UWAGA:** Zasilanie na listwie można włączyć dopiero PO poprawnym przygotowaniu stanowiska do pracy, oraz PO zweryfikowaniu przez prowadzącego zajęcia!

**UWAGA:** Nie wolno odłączać adaptera RPi, wentylatora, oraz konwertera USB <-> UART. Urządzenia należy podłączyć w następujący sposób (**zasilanie musi być wyłączone!**):



Podłączenie UART (powinno być tak podłączone):



Następnie należy przygotować stanowisko do pracy: umieścić urządzenia w dostępnym miejscu, w taki sposób, żeby nie przeszkadzały w korzystaniu z klawiatury. Przewody należy przeciągnąć w taki sposób, aby nie przeszkadzały w korzystaniu z myszy i klawiatury, oraz żeby był zapewniony dostęp dołącznika listwy oraz urządzenia RPi.

**UWAGA:** Po przygotowaniu stanowiska do pracy, przed włączeniem zasilania, należy zgłosić się do prowadzącego, aby sprawdził poprawność połączeń.

## Pierwsze uruchomienie RPi

Przed włączeniem zasilania RPi, na komputerze host należy podłączyć się do terminala UART jednym z dwóch programów:

1. tio - uruchamiamy poleceniem:

```
tio /dev/ttyUSB0
```

Po podłączeniu przez tio powinien wyświetlić się status "connected".

Aby zamknąć terminal tio należy nacisnąć kombinację klawiszy "CTRL+t q", czyli: wciskamy jednocześnie CTRL+t, a następnie po zwolnieniu należy wcisnąć klawisz "q".

2. lub minicom - uruchamiamy poleceniem:

```
minicom -D /dev/ttyUSB0
```

W przypadku minicom, trzeba zmienić konfigurację - wyłączyć sprzętową kontrolę przepływu:

"*CTRL+a o*" - menu konfiguracji portu szeregowego, a następnie wybieramy:

*serial port setup -> hardware flow control -> No*

następnie należy zachować konfigurację:

*Save setup as dfl*

Można też zmieniać konfigurację minicoma (wg własnego uznania):

"*CTRL+a z*" - menu konfiguracji minicoma

Po włączeniu zasilania, na terminalu UART na host zobaczymy logi z bootloadera, a następnie logi z bootowania kernela.

Jeśli nie zostanie wciśnięty przycisk SW4 na adapterze RPi podczas bootowania, to załadowany zostanie system ratunkowy Buildroot.

W takim przypadku pojawią się logi z Linuxa i znak zachęty:

*Welcome to Buildroot rescue OS  
rescue login:*

Należy podać następujące dane do logowania:

login: *root*

hasła nie ma.

Po zalogowaniu należy uruchomić ręcznie DHCP i poczekać na przyznanie adresu IP:

**udhcpc**

Można sprawdzić stan połączenia sieciowego na RPi:

- wyświetlenie m.in. IP RPi:

**ifconfig**

- pingowanie komputera host:

**ping <adres ip hosta>**

i analogicznie na host (PC):

- wyświetlenie m.in. IP hosta (PC):

**ifconfig**

- pingowanie RPi:

**ping <adres ip RPi>**

Adres IP hosta to zwykle 10.42.0.1, a RPi otrzymuje adres 10.42.0.X

## Kopiowanie plików na RPi

Na komputerze host można postawić serwer HTTP za pomocą polecenia:

**python3 -m http.server**

Powyższe polecenie udostępnii przez HTTP na porcie 8000 zawartość katalogu roboczego (czyli tego, w którym polecenie zostało wykonane).

Aby ściągnąć plik udostępniony w ten sposób, należy na RPi wykonać polecenie:

**wget http://<adres\_ip\_hosta>:8000/<nazwa\_pliku>**

przy czym <nazwa\_pliku> to ścieżka do pliku względem katalogu roboczego (dla którego został uruchomiony serwer HTTP).

Drugim sposobem jest kopiowanie plików za pomocą ssh (jeśli ssh jest dostępne, czyli lokalnie jest zainstalowany klient ssh, a na drugiej maszynie działa serwer ssh), poleceniem scp, np.:

- ściągnięcie pliku:  
`scp <user_name>@<src_ip>:<src_filename> dst_filename`
- wysłanie pliku:  
`scp src_filename <user_name>@<dst_ip>:<dst_filename>`

## Logowanie za pomocą ssh systemie ratunkowym na RPi

W systemie ratunkowym ssh jest włączony, ale mogą wystąpić problemy z logowaniem ze względu na konfigurację konta użytkownika root na RPi. Jeśli wystąpią problemy z logowaniem, np. pojawi się komunikat “Permission denied, please try again.”, to można spróbować zmienić hasło na systemie ratunkowym poleceniem `passwd`

## Kompilacja obrazu Linuxa w Buildroot

### Obraz dla Raspberry Pi 4B z initramfs

Budujemy Linux, zgodnie z wykładem 1 i 2.

Ściągamy buildroot:

<https://buildroot.org/downloads/buildroot-2021.08.tar.bz2>

Rozpakowujemy archiwum w katalogu, w którym zamierzamy go zbudować (na komputerze host).

**UWAGA:** zawartość katalogu domowego na komputerach w P113 i P114 będzie usuwana po każdym zajęciach.

Można spakować zbudowany Buildroot, np.:

`tar -cf buildroot.tar buildroot-2021.08`

oraz zachować taką spakowaną paczkę np. w chmurze lub na pendrive. Proszę nie wgrywać całego buildroota na gitlab! Na gitlabie przechowujemy tylko konfigurację (plik .config).

Aby zbudować Buildroot wykonujemy polecenia (w katalogu z rozpakowaną paczką buildroot-2021.08.tar.bz2):

`make raspberrypi4_64_defconfig`

`make menuconfig`

W menu zaznaczamy

*Toolchain --> Toolchain type: External toolchain*

Należy także włączyć initramfs i wyłączyć ext2/3/4.

Proszę pamiętać o włączeniu kompresji obrazu (w menuconfig buildroota).

W przypadku błędów związanych z brakiem miejsca na obrazie karty SD (na końcu budowania Buildroota), należy:

- a) sprawdzić, czy jest włączona kompresja
- b) zwiększyć rozmiar partycji boot, w budowanym obrazie karty SD, w pliku:  
`<buildroot>/board/raspberrypi4-64/genimage-raspberrypi4-64.cfg`

Plik `<buildroot>/board/raspberrypi4-64/genimage-raspberrypi4-64.cfg` ma strukturę json.

Zwiększenie rozmiaru partycji boot:

```
image boot.vfat {  
    ...  
    size = <nowy_rozmiar_w_MB>M  
}  
...
```

Rozmiar powinien być wielokrotnością liczby 4 (aby zachować alignment 4MB).

Na końcu budujemy obraz polecienniem:

```
make
```

Budowanie zajmuje trochę czasu.

Zbudowany obraz Linuxa z initramfs znajduje się w pliku

```
<buildroot>/output/images/Image
```

Aby uruchomić obraz należy skorzystać z metod opisanych w [Uruchomienie zbudowanego obrazu](#).

## Uruchomienie zbudowanego obrazu

Obraz (plik `Image`), wraz z linią poleceń (plik `cmdline.txt`) i drzewem urządzeń (plik `bcm2711-rpi-4-b.dt`) można pobrać przez http z poziomu systemu ratunkowego i zapisać na partycji 1 (boot) w katalogu `/user`.

Należy skopiować ww. pliki z host na RPi, korzystając z metod opisanych w [Kopiowanie plików na RPi](#).

Należy zamontować partycję 1 (boot) karty SD, np. w katalogu `/mnt`, za pomocą polecenia:  
`mount /dev/mmcblk0p1 /mnt`

Następnie kopujemy pliki `Image`, `cmdline.txt` i `bcm2711-rpi-4-b.dt` do katalogu `/mnt/user`. Należy pamiętać, aby plik `Image` zapisać pod nazwą `kernel8.img`

Zrestartuj RPI:

```
reboot
```

i przytrzymaj wciśnięty przycisk SW4 tak długo, aż zacznie się bootować kernel.

## Obraz dla Raspberry Pi 4B bez initramfs

Można posłużyć się zbudowanym w poprzednim ćwiczeniu Buildrootem, na bazie poprzedniej konfiguracji (proszę zapisać kopię pliku .config przed dokonaniem zmian - będzie potrzebna do raportu!).

Trzeba najpierw usunąć poprzedni obraz polecienniem:

```
make linux-dirclean
```

Następnie w

```
make menuconfig
```

powinno być zaznaczone:

*Toolchain --> Toolchain type: External toolchain*

oraz powinna być zaznaczona kompresja obrazu kernela (z poprzedniego ćwiczenia).

Należy wyłączyć initial RAM filesystem (initramfs) oraz włączyć wsparcie dla ext2/3/4.

**UWAGA:** w przypadku ustawienia zbyt małego rozmiaru systemu plików ext2 może pojawić się błąd podczas komplikacji.

Budujemy obraz polecienniem:

```
make
```

Wynikowy obraz (plik <buildroot>/<output>/<images>/Image) powinien być mniejszy niż w poprzednim ćwiczeniu (które wie dlaczego?).

Dodatkowo, interesuje nas plik z systemem plików:

<buildroot>/output/images/rootfs.ext2

który należy nagrać na partycji 2 (rootfs) na karcie SD w RPi, z poziomu systemu ratunkowego.

Należy przekopiować plik rootfs.ext2 z hosta na RPi, korzystając z metod opisanych w [Kopiowanie plików na RPi](#), a następnie należy nagrać system plików z poziomu systemu ratunkowego na RPi na kartę SD, na partycję 2 (rootfs):

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

Aby uruchomić obraz należy skorzystać z metod opisanych w [Uruchomienie zbudowanego obrazu](#).

Na końcu należy sprawdzić, czy załadowany system rzeczywiście korzysta z systemu plików na karcie SD. Najprostszy test polega na utworzeniu pliku w katalogu głównym, np.:

```
touch /test.txt
```

a następnie należy uruchomić system zbudowany w Buildroot ponownie i sprawdzić, czy ten plik dalej istnieje (powinien przetrwać reboot).

## Zadanie domowe: Obraz na 64-bitową maszynę wirtualną “Virt” i uruchomienie w QEMU

W świeżo rozpakowanym Buildroocie proszę skonfigurować zbudowanie obrazu systemu Linux dla 64-bitowej maszyny Virt.

W konfiguracji proszę ustawić nazwę systemu na Nazwisko1\_Nazwisko2 (nazwiska obu członków zespołu).

1. Proszę skompilować wersję używającą ramdysku startowego i uruchomić ją w QEMU
2. Proszę skompilować wersję używającą rzeczywistego systemu plików i uruchomić ją w QEMU