Systemy komputerowe w sterowaniu i pomiarach (SKPS)
Instrukcja do laboratorium 1

dr hab. inż. Wojciech Zabołotny, mgr inż. Dawid Seredyński
Ostatnia aktualizacja: 03.03.2021

Informacje wstępne

Cel laboratorium

Celem laboratorium jest zapoznanie się z Buildroot oraz uruchomienie go w qemu i na RPi. Laboratorium 1 jest za 6 punktów, w tym:

- 0 - zadanie domowe - bez punktacji, ale zrobienie zadania jest bardzo pomocne w pracy na laboratorium
- 2 - wejściówka
- 4 - 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 wejściówki

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 4 partycje:

1. **boot** - zawiera:
    a. pliki *.dtb - DT - device tree, drzewo urządzeń,
    b. katalog overlays - overlays dla DT
    c. config.txt - plik z konfiguracją dla bootloadera
    d. u-boot.bin - obraz U-boot
    e. boot.scr.uimg - skrypt dla U-boot, który uruchamia Raspberry Pi OS
    f. start.elf - firmware GPU
    g. fixup.dat - plik związany z firmware
    h. cmdline.txt - plik z linią poleceń dla systemu ratunkowego
    i. kernel8.img - oryginalny Raspberry Pi OS (64-bit) dla RPi 4B, system ratunkowy
2. **rootfs_recovery** - system plików dla systemu ratunkowego
3. **images** - zawiera obrazy załadowane przez sieć, które zostały zbudowane w ramach zajęć
4. **rootfs** - system plików dla załadowanego obrazu

Studenci mogą zmieniać tylko zawartość partycji nr 3 (images) oraz 4 (rootfs). Zmiany partycji 2 (rootfs_recovery) także są dopuszczalne i zachodzą podczas korzystania z systemu ratunkowego.

Zestaw laboratoryjny

Preferujemy restartowanie RPi za pomocą polecenia `reboot` (w Linuxie) lub `reset` (w U-boot). 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 zwinąć i spiąć kable, spakować elementy do pudełek i opakowań.

Kody źródłowe

Kody źródłowe i pliki konfiguracyjne przechowujemy na wydziałowym gitlabie:
https://gitlab-stud.elka.pw.edu.pl

Nazwa repozytorium:
skps21z_inazwisko1_inazwisko2
(inazwisko - imię i nazwisko, bez polskich liter)

Sugerowana struktura katalogów w repozytorium:

- cw1
  - BR_RPi
    - .config
  - BR_virt
    - .config
- cw2
  - ...
- ...

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

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 /

Polecenia, które są wpisywane w konsoli **U-boot** oznaczone są czarnym tłem i białą czcionką, np.:
printenv bootargs

Przygotowanie stanowiska

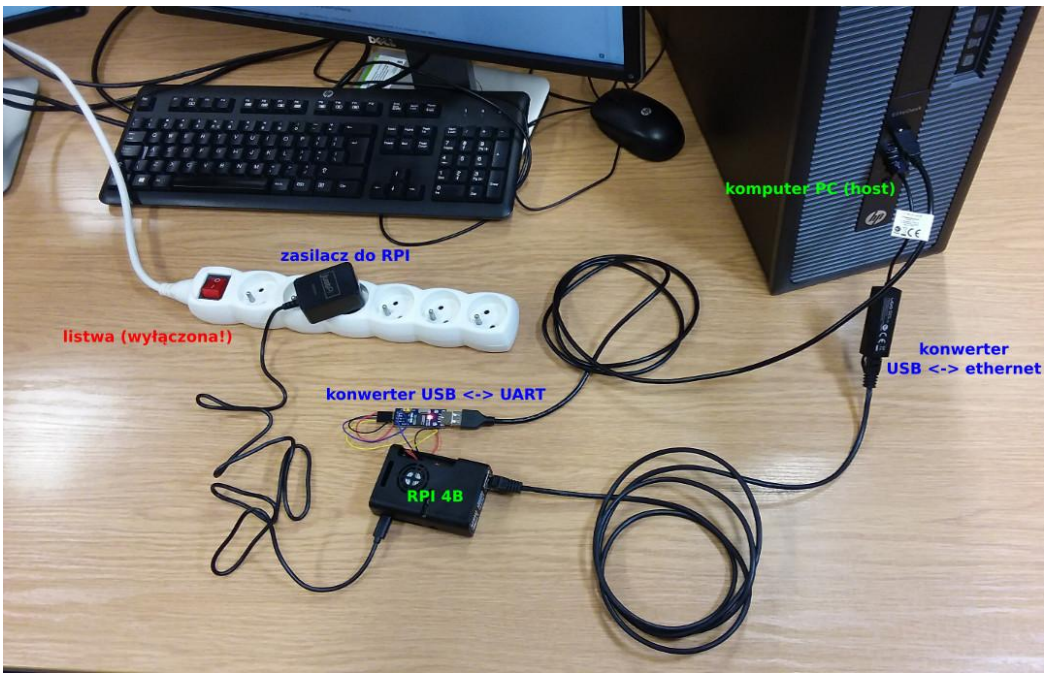
Studenci samodzielnie przygotowują stanowisko, podłączając urządzenia według instrukcji, ale **przed jego uruchomieniem (włączeniem zasilania) muszą skonsultować się z prowadzącym**.

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

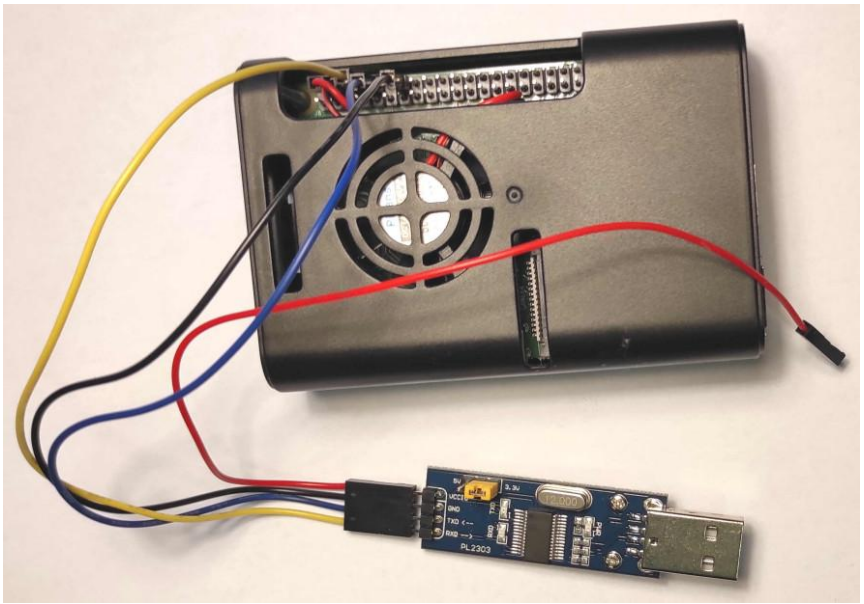
Podstawowy zestaw do laboratorium wygląda następująco:

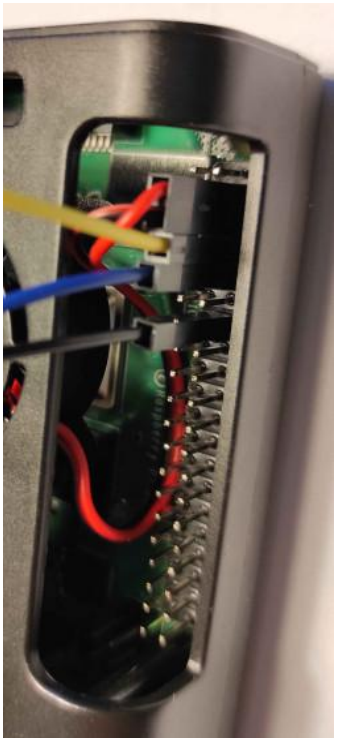
UWAGA: Zasilanie na listwie można włączyć dopiero PO poprawnym przygotowaniu stanowiska do pracy,oraz PO zweryfikowaniu przez prowadzącego zajęcia!

Urządzenia należy podłączyć w następujący sposób (**zasilanie musi być wyłączone!**):

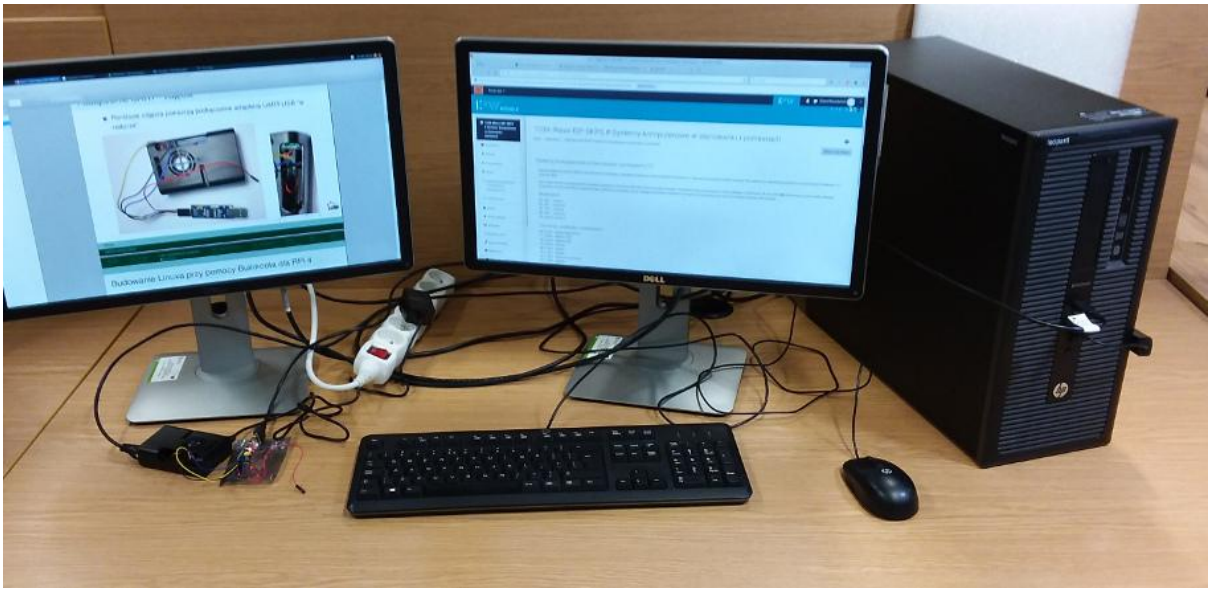


Podłączenie UART:





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 do włącznika 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 dlf
   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 U-boot.
Jeśli nie zostanie wciśnięty klawisz podczas 2s po uruchomieniu U-boot, to załadowany zostanie system ratunkowy Raspberry Pi OS.

W takim przypadku pojawią się logi z Linuxa i znak zachęty:
raspberrypi login:
Należy podać następujące dane do logowania:
login: *pi*
hasło: *raspberry*

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 udostępni 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_fileneme <user_name>@<dst_ip>:<dst_filename>

Kompilacja obrazu Linuxa w Buildroot

Obraz dla Raspberry Pi 4B z initramfs

Budujemy od razu Linux i u-boot, 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.

UWAGA: zawartość katalogu domowego na komputerach w P113 będzie usuwana po każdych zajęciach.

Można spakować zbudowany Buildroot, np.:
`tar -cf buildroot.tar buildroot-2021.08`
oraz zachować taką spakowaną paczkę np. w chmurze lub na pendrive

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 poleceniem:
`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

Z karty SD, z użyciem systemu ratunkowego

Obraz można pobrać przez http i zapisać na partycji 3 (images) z poziomu systemu ratunkowego.
Następnie można uruchomić pobrany obraz w U-boot bezpośrednio z partycji.

Należy skopiować plik z obrazem z host na RPi, korzystając z metod opisanych w Kopiowanie plików na RPi.

Przenieś plik z obrazem (*Image*) do katalogu */images*, w którym jest zamontowana partycja nr 3 karty SD (partycja images typu FAT):
```
sudo mv Image /images/
```

Zrestartuj RPI:
```
sudo reboot -h now
```

Należy uważnie obserwować logi, które pojawiają się na konsoli UART. Kiedy pojawi się napis:
Hit any key to stop autoboot:
Należy wcisnąć ENTER.
U-boot wejdzie w tryb interaktywny.

Poniższe polecenia wykonujemy w konsoli U-boot.
Załaduj z partycji typu *FAT* (*fatload*) na karcie *SD* (*mmc*) o numerze *0* z partycji nr *3* pod adres *${kernel_addr_r}* plik o nazwie *Image*
```
fatload mmc 0:3 ${kernel_addr_r} Image
```

UWAGA: Jeśli korzystamy z rootfs (nie initramfs), to należy dodać argument *root=/dev/mmcblk0p4*
```
setenv bootargs console=tty1 console=ttyAMA0,115200 root=/dev/mmcblk0p4 rootfstype=ext4 rootwait
```

Uruchom obraz Linuxa (może być skompresowany lub nie), który jest załadowany pod adresem *${kernel_addr_r}*, bez initrd, z drzewem urządzeń pod adresem *${fdt_addr}*:
```
booti ${kernel_addr_r}  - ${fdt_addr}
```

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 poleceniem:
`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 kompilacji.
Budujemy obraz poleceniem:
`make`

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

Dodatkowo, interesuje nas plik z systemem plików:
<buildroot>/output/images/rootfs.ext2
który należy nagrać na partycji 4 na karcie SD w RPi, z poziomu systemu ratunkowego.

Należy przekopiować plik *rootfs.ext2* z host na RPi, korzystając z metod opisanych w Kopiowanie plików na RPi, a następnie należy nagrać system plików z poziomu systemu ratunkowego na RPi:
`sudo dd if=rootfs.ext2 of=/dev/mmcblk0p4 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