

INSTRUKCJA

1. Przygotowanie płytki

- połączyć płytkę kablem typu Ethernet z komputerem
- połączyć płytkę kablem szeregowym z portem COM1 w komputerze. Port COM1 jest portem znajdującym się po prawej stronie (patrząc od tyłu obudowy komputera)
- włożyć kartę microSD do złącza znajdującego się na płycie
- podłączyć zasilacz
- włączyć płytkę, po kilkunastu sekundach na ekranie powinny pokazać się komunikaty systemu
- zapoznać się z instrukcją do płytki Devkit8500, plik z instrukcją zostanie udostępniony przez prowadzącego

2. Przygotowanie maszyny wirtualnej

- uruchomić program Oracle VirtualBox
- utworzyć nową maszynę wirtualną typu Ubuntu 32-bit o domyślnych parametrach pamięci. Jako dysk wybrać plik wskazany przez prowadzącego
- otworzyć ustawienia sieciowe (Network) maszyny wirtualnej. Włączyć dwa adaptory sieciowe i zmienić ich tryb na "Bridged". Pierwszy adapter powinien być zmostkowany z Realtek PCIe GBE Family Controller a drugi z Realtek PCI GBE Family Controller.
- otworzyć ustawienia portów szeregowych (Serial Ports) maszyny wirtualnej. Włączyć port COM1 i ustawić go w trybie Host Device. W polu Port/File path wpisać "COM1:" (bez cudzysłowu).
- otworzyć ustawienia współdzielonych folderów (shared folders), utworzyć nowy folder współdzielony. Jako ścieżkę podać D:/VM_Ubuntu/share (jeżeli taki folder nie istnieje, należy go utworzyć). Jako nazwę podać share.
- uruchomić maszynę wirtualną i zalogować się do systemu (login: dmcs, hasło: dmcs)
- sprawdzić ustawienia sieci/IP komendą ifconfig. Jeden interfejs sieciowy powinien mieć przydzielony adres IP otrzymany z serwera DHCP. Interfejs ten będzie służył do połączenia z Internetem. Do połączenia z płytką będzie służył drugi interfejs, należy odczytać jego nazwę (eth*). Następnie edytować plik konfiguracyjny sieci systemu Ubuntu poleceniem `sudo nano /etc/network/interfaces`. Podmienić obecny w pliku interfejs na ten, który wyświetlił się przy uruchomieniu ifconfig. Na przykład, jeżeli ifconfig wyświetlił interfejs eth7, plik /etc/network/interfaces powinien mieć postać

```
auto lo
iface lo inet loopback
auto eth7
allow-hotplug eth7
iface eth7 inet static
address 192.168.0.1
netmask 255.255.255.0
```
- Zrestartować proces zarządzający siecią:
`sudo /etc/init.d/networking restart`

- Sprawdzić poleceniem ifconfig czy na interfejsie jest poprawny adres IP (192.168.0.2)
- użyć polecenia ping 192.168.0.2 do sprawdzenia poprawności połączenia maszyny wirtualnej z płytką
- połączyć się z płytką za pomocą ssh, z użyciem polecenia ssh devkit (devkit jest aliasem na root@192.168.0.2)
- w przypadku zapytania o hasło wcisnąć enter

- do transferu plików między systemem Windows i maszyną wirtualną można wykorzystać folder współdzielony. W tym celu należy stworzyć folder o nazwie share w katalogu domowym (komenda cd && mkdir share) a następnie zamontować folder współdzielony (sudo mount -t vboxsf share ~/share/). Przetestować działanie folderu współdzielonego.

3. Sprawdzenie działania konsoli

- do odczytu danych z portu szeregowego na maszynie wirtualnej służy program minicom.
- zapoznać się z opcjami programu za pomocą polecenia minicom -h
- uruchomić odczyt z portu COM1 z(w Ubuntu widziany jako /dev/ttyS0) z prędkością 115200 bodów używając komendy sudo minicom -D /dev/ttyS0 -b 115200
- wcisnąć przycisk reset na płytce, w oknie programu minicom powinny pojawić się komunikaty uruchamiających się bootloadera i kernela

4. Analiza procesu uruchamiania systemu na płytce

-
- przeanalizować zawartość pierwszej partycji (FAT) karty SD
- proces bootowania jest przedstawiony na rys xxx.
- przeanalizować zawartość pliku boot.scr
- włożyć kartę do płytki i uruchomić odczyt z konsoli programem minicom. Uruchomić płytke i przerwać bootowanie w momencie wyświetlenia komunikatu "Press any key to stop autoboot", co spowoduje wejście w konsolę bootloadera u-boot. Zapoznać się z komendami bootloadera poleceniem help. Wyświetlić aktualne ustawienia bootloadera poleceniem printenv. Przenalizować proces bootowania systemu (na podstawie zmiennej środowiskowej bootcmd). Porównać zawartość konfiguracji u-boota ze skrypcem u-boota podanym przez prowadzącego. Przeanalizować znaczenie parametrów takich jak console, root, bootargs itp. oraz komend fatload, bootm, itp.

5. Kompilacja bootloaderów oraz jądra systemu linux.

- przenieść zawartość partycji FAT karty SD do katalogu domowego (kasując zawartość partycji FAT na karcie)
- zgodnie z instrukcją do płytki Devkit8500 skompilować kolejno bootloader x-loader, bootloader u-boot oraz jądro systemu linux. W przypadku wystąpienia komunikatów o błędzie (no rule to make target) przeanalizować pliki makefile i odpowiednio zmodyfikować komendę kompilacji.

Wgrać pliki wynikowe na kartę. Plik skryptu u-boota zostanie podany przez prowadzącego, należy zamienić go programem mkimage (zgodnie z instrukcją do

płytki Devkit8500) na plik binarny i również umieścić na karcie SD. Uwaga: tool mkimage nie działa poprawnie wewnątrz folderu współdzielonego, przy generowaniu plik skryptu u-boota należy skorzystać z innego folderu. Sprawdzić czy płytka z nową zawartością karty uruchamia się poprawnie.

Zgodnie z instrukcją do płytki Devkit8500 wgrać pliki startowe do pamięci NAND. Przetestować, czy płytka uruchamia się poprawnie z pamięci NAND.

6. Znaleźć w internecie informacje na temat bootowania systemu plików po sieci (z wykorzystaniem protokołu NFS) - słowa kluczowe: NFS, u-boot, script. Zmodyfikować skrypt w ten sposób, żeby system plików był ładowany z maszyny wirtualnej, przy czym system plików znajduje się w katalogu /home/dmcs/targetfs. Sprawdzić, czy płytka zbootuje się poprawnie z nowymi ustawieniami. Wejść do katalogu /home/dmcs/targetfs/home/root na maszynie wirtualnej i utworzyć tam plik o dowolnej nazwie. Następnie połączyć się po ssh z płytką i sprawdzić czy w katalogu /home/root znajduje się utworzony plik.

7. Wrócić do poprzedniej wersji skryptu (bootowanie systemu plików z partycji na karcie SD). Następnie połączyć się po ssh z płytką oraz przejść do folderu z przykładowymi programami przygotowanymi przez Texas Instruments (/usr/share/ti). Uruchomić następujące przykłady:

- dekodowanie filmu: katalog dvSDK-demos, polecenie ./decode -O lcd -v ../data/videos/davincieffect.264
- obsługa kamery: katalog linux-driver-examples, polecenie saMmapLoopback -s 1
- wykorzystanie biblioteki c6accel: katalog c6accel, polecenie ./loadmodules*.sh oraz ./c6accel_app
- wykorzystanie biblioteki c6run: katalog c6run, polecenie ./unloadmodules.sh, ./loadmodules.sh co pozwala na uruchomienie przykładów znajdujących się w folderze examples

8. Kompilacja prostego programu.

- przeanalizować podany przez prowadzącego przykładowy plik Makefile
- z katalogu dvSDK/psp/linux-driver-examples/video skopiować program saVideoARGB.c do katalogu domowego (np. /home/dmcs/work)
- przeanalizować program zaczynając od funkcji main. Odszukać nazwy dwóch plików obrazów otwieranych przez program i skopiować je do katalogu work.
- odpowiednio zmienić Makefile tak, aby poprawnie kompilował się skopiowany plik (można zmienić też nazwę pliku na prostszą, np. rgb.c)
- skopiować pliki obrazów na płytke
- uruchomić program na płytce i przeanalizować jego działanie. Czy obrazy wyświetlają się poprawnie?
- otworzyć na maszynie wirtualnej program gimp. W programie otworzyć plik obrazu "omap.rgb" zaznaczając jako typ pliku RAW input data. Zaobserwować różnice w obrazie wyświetlanym w programie gimp i wyświetlanym na płytce.
- zmodyfikować program w taki sposób, żeby obraz wyświetlany był poprawnie. Zamiast obrazu omap.rgb można użyć innego obrazu podanego przez prowadzącego. Wskazówka: przeanalizować dokładnie pętlę w programie, która kopiuje bajty z pliku do tablicy w pamięci.
- jeżeli obraz wyswietla się poprawnie, spróbować zmodyfikować wyświetlany obraz

poprzez dodanie do składowej G wartości 50. Należy pamiętać o zabezpieczeniu przed przepełnieniem się zmiennej (nie może ona być większa niż 255).

9. Z katalogu `dvSDK/psp/linux-driver-examples/video` skopiować program `saMmapLoopback.c` do katalogu domowego (np. `/home/dmcs/work`). Można zmienić jego nazwę na prostszą np. `camera.c`. Skompilować program i uruchomić go z parametrem `-s 1` (`./camera -s 1`). W przypadku gdy na ekranie nie wyświetla się poprawnie obraz z kamery poprosić o pomoc prowadzącego.

- w programie zidentyfikować wskaźnik to ramek przychodzących z kamery oraz wskaźnik do ramek wysyłanych na wyświetlacz.
- odnaleźć fragment kodu odpowiedzialny za kopiowanie ramek
- wyszukać w internecie informacje o formacie YUV oraz jego podtypie YUYV. Dopisać do programu prostą pętlę, która będzie usuwała z obrazu składowe U i V, tak aby powstał obraz w skali szarości.
- wyszukać w internecie informacje o filtrze dolnoprzepustowym dla obrazu cyfrowego, a następnie dopisać do programu funkcję, która stosuje taki filtr dla obrazu w formacie YUYV. Funkcja powinna przyjmować jako parametry wskaźnik do tablicy wejściowej, wskaźnik do tablicy wyjściowej oraz wysokość/szerokość obrazu. Filtr dolnoprzepustowy powinien zostać zrealizowany z wykorzystaniem maski $[3][3]=\{1,1,1,1,1,1,1,1,1\}$. Sprawdzić działanie filtru na obrazie z kamery. Wywołać funkcję kilkakrotnie (np. pięć razy) i zaobserwować zmiany w obrazie oraz szybkość działania programu (zmianę ilości klatek na sekundę).
- wyszukać w internecie informacje o filtrze krawędziowym Sobela dla obrazu cyfrowego, a następnie dopisać do programu funkcję, która stosuje taki filtr dla obrazu w formacie YUYV. Funkcja powinna przyjmować jako parametry wskaźnik do tablicy wejściowej, wskaźnik do tablicy wyjściowej oraz wysokość/szerokość obrazu. Filtr Sobela powinien zostać zrealizowany z wykorzystaniem maski $[3][3]=\{-1,-1,-1,-1,8,-1,-1,-1,-1\}$. Sprawdzić działanie filtru na obrazie z kamery.
- wyszukać w internecie informacje o operacji dylatacji i erozji. Zaimplementować funkcje realizujące obie operacje (każdy przetworzony piksel powinien przyjmować wartość minimalną/maksymalną z ośmiu wartości pikseli sąsiednich i swojej poprzedniej wartości).
- sprawdzić działanie powyższych operacji dla usuwania zakłóceń z obrazu (jedna operacja erozji, jedna operacja dylatacji)
- wywołać powyższe funkcje kilkakrotnie (np. 3 operacje erozji, potem 3 operacje dylatacji), zaobserwować jak zmienił się obraz wyjściowy oraz jak zmieniła się szybkość działania programu.