Hands-on Linux Academy: interfejsy komunikacyjne

Łukasz Skalski <contact@lukasz-skalski.com>

Warszawa, 07-11-2017

Plan szkolenia

ĆWICZENIE 1: *Przygotowanie karty SD z systemem dla komputera VisionSOM-6ULL*

- Instrukcja przygotowania karty SD w systemach operacyjnych Linux i Windows
- Linux w systemach wbudowanych? Czy to ma sens?
- Uruchomienie płytki *VisionSOM-6ULL* oraz konfiguracja sieci (serwer DHCP)

Plan szkolenia

ĆWICZENIE 1: *Przygotowanie karty SD z systemem dla komputera VisionSOM-6ULL*

- Instrukcja przygotowania karty SD w systemach operacyjnych Linux i Windows
- Linux w systemach wbudowanych? Czy to ma sens?
- Uruchomienie płytki *VisionSOM-6ULL* oraz konfiguracja sieci (serwer DHCP)

ĆWICZENIE 2: *Interfejsy komunikacyjne w przykładach*

- Sterowanie portami GPIO poprzez interfejs */sys/class/gpio*
- Obsługa przycisków z wykorzystaniem podsystemu *GPIO Buttons*
- Obsługa diod LED z wykorzystaniem podsystemu *LED Class Driver*
- Magistrala SPI na przykładzie aplikacji typu "*loopback*"
- Magistrala I2C na przykładzie obsługi modułu żyroskopu

Plan szkolenia

ĆWICZENIE 1: *Przygotowanie karty SD z systemem dla komputera VisionSOM-6ULL*

- Instrukcja przygotowania karty SD w systemach operacyjnych Linux i Windows
- Linux w systemach wbudowanych? Czy to ma sens?
- Uruchomienie płytki *VisionSOM-6ULL* oraz konfiguracja sieci (serwer DHCP)

ĆWICZENIE 2: *Interfejsy komunikacyjne w przykładach*

- Sterowanie portami GPIO poprzez interfejs */sys/class/gpio*
- Obsługa przycisków z wykorzystaniem podsystemu *GPIO Buttons*
- Obsługa diod LED z wykorzystaniem podsystemu *LED Class Driver*
- Magistrala SPI na przykładzie aplikacji typu "*loopback*"
- Magistrala I2C na przykładzie obsługi modułu żyroskopu

ĆWICZENIE 4: *Time-to-market* w systemach wbudowanych, czyli wykorzystanie gotowych komponentów oprogramowania

- *Node.js* - systemy wbudowane i JavaScript?
- Prosta implementacja serwera WWW z wykorzystaniem *Node.js*
- Serwer WWW z podziałem na funkcje *front-end* oraz *back-end*
- Komunikacja *front-end*<->*back-end* z wykorzystaniem *socket.io*
- Serwer WWW z odczytem danych z modułu żyroskopu
- Rozbudowa interfejsu serwera WWW o elementy grafiki 3D (*Three.js*)

ĆWICZENIE 1:

Przygotowanie karty SD z systemem dla komputera VisionSOM-6ULL

Przygotowanie karty SD w systemie Linux

- Jedną najprostszych metod zapisania obrazu na karcie SD jest wykorzystanie narzędzia `dd`:

```
dd if=<pliku_wejściowy> of=<plik_wyjściowy> <dodatkowe opcje>
```

Przygotowanie karty SD w systemie Linux

- Jedną najprostszych metod zapisania obrazu na karcie SD jest wykorzystanie narzędzia `dd`:

```
dd if=<pliku_wejściowy> of=<plik_wyjściowy> <dodatkowe opcje>
```

- Poprawne określenie pliku wyjściowego reprezentującego podłączoną do czytnika kartę SD, pozwala na ostateczne określenie formy polecenia `dd`:

```
sudo dd if=/path/to/linux-academy.img of=/dev/sdX bs=4M oflag=dsync
```

Przygotowanie karty SD w systemie Linux

- Jedną najprostszych metod zapisania obrazu na karcie SD jest wykorzystanie narzędzia `dd`:

```
dd if=<pliku_wejściowy> of=<plik_wyjściowy> <dodatkowe opcje>
```

- Poprawne określenie pliku wyjściowego reprezentującego podłączoną do czytnika kartę SD, pozwala na ostateczne określenie formy polecenia `dd`:

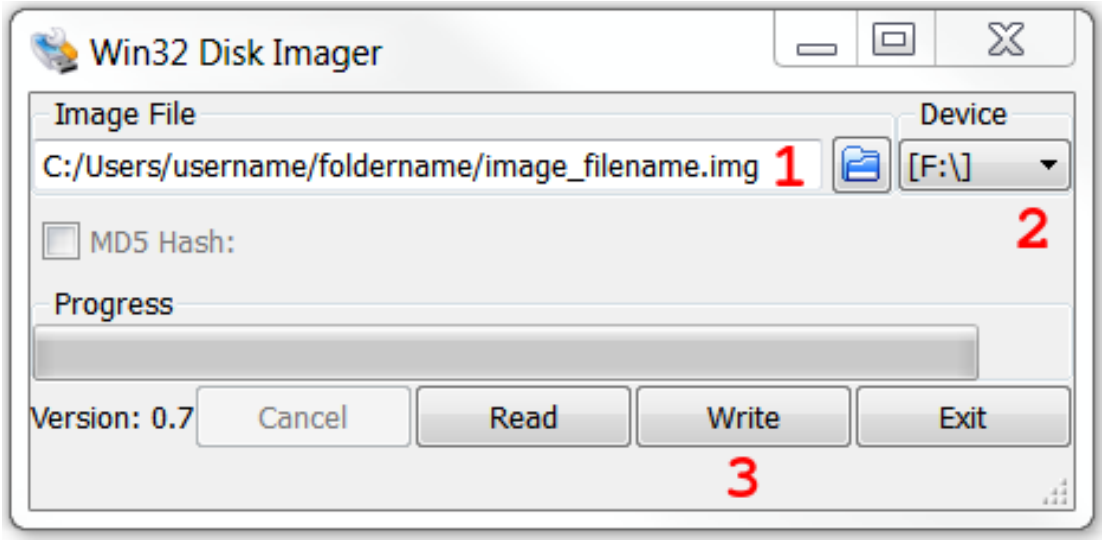
```
sudo dd if=/path/to/linux-academy.img of=/dev/sdX bs=4M oflag=dsync
```

- Polecenie `dd` nie umożliwia nam monitorowania postępu zapisu danych na kartę SD. Wygodnym rozwiązaniem jest wykorzystanie narzędzia `pv`, który wyświetla informacje o postępie odczytu danych z pliku:

```
pv linux-academy.img | dd of=/dev/zero bs=4M ofag=dsync
25.5MiB 0:00:02 [5.03MiB/s] [====>                ] 21% ETA 0:00:27
```

Przygotowanie karty SD w systemie Windows

Jedną z najprostszych opcji w przygotowaniu karty SD w systemie operacyjnym Windows jest wykorzystanie darmowego narzędzia `Win32DiskImager`:



Po umieszczeniu czytnika z kartą w slocie USB, użytkownik zostanie poproszony o:

- wskazanie ścieżki do pliku **.img* z obrazem systemu (1),
- wskazanie urządzenia docelowego (2),
- przeprowadzenie operacji zapisu poprzez wybranie przycisku *Write* (3).

Linux w systemach wbudowanych?

System operacyjny definiowany jako:

- Menedżer zasobów

Linux w systemach wbudowanych?

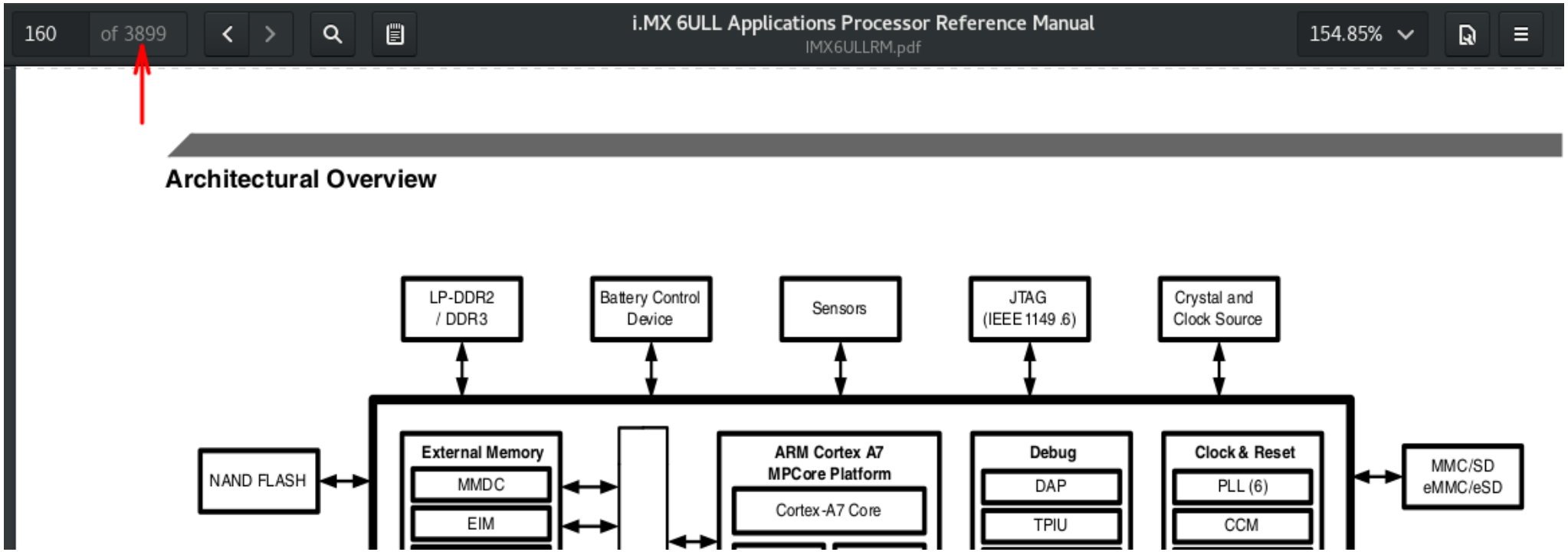
System operacyjny definiowany jako:

- Menedżer zasobów
- Abstrakcja sprzętu

Linux w systemach wbudowanych?

System operacyjny definiowany jako:

- Menedżer zasobów
- Abstrakcja sprzętu



Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

- od projektowanego urządzenia wymagana jest wysoka niezawodność działania,

Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

- od projektowanego urządzenia wymagana jest wysoka niezawodność działania,
- urządzenie ma posiadać rozbudowany graficzny interfejs użytkownika, przetwarzać duże ilości danych, … ,

Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

- od projektowanego urządzenia wymagana jest wysoka niezawodność działania,
- urządzenie ma posiadać rozbudowany graficzny interfejs użytkownika, przetwarzać duże ilości danych, … ,
- programowanie w języku C/C++ nie jest naszą najmocniejszą stroną,

Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

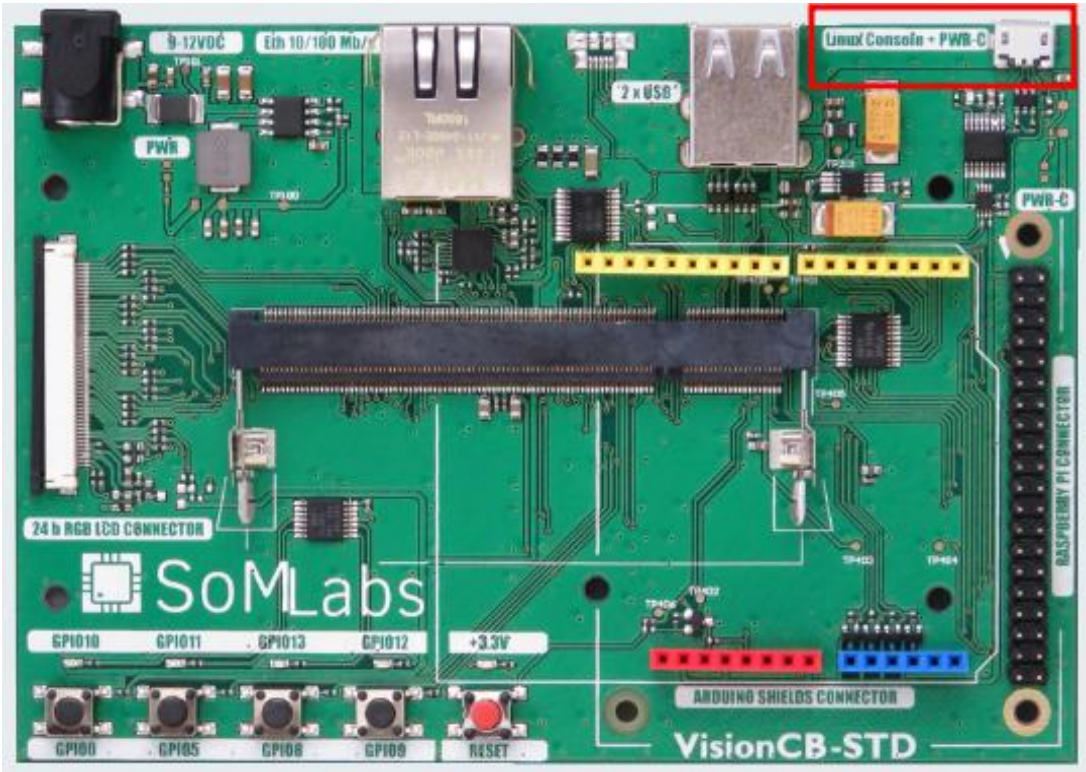
- od projektowanego urządzenia wymagana jest wysoka niezawodność działania,
- urządzenie ma posiadać rozbudowany graficzny interfejs użytkownika, przetwarzać duże ilości danych, … ,
- programowanie w języku C/C++ nie jest naszą najmocniejszą stroną,
- projektowane urządzenia wymaga implementacji stosów USB, Ethernet, Bluetooth,

Linux w systemach wbudowanych?

System operacyjny Linux jest odpowiednim rozwiązaniem jeżeli:

- od projektowanego urządzenia wymagana jest wysoka niezawodność działania,
- urządzenie ma posiadać rozbudowany graficzny interfejs użytkownika, przetwarzać duże ilości danych, … ,
- programowanie w języku C/C++ nie jest naszą najmocniejszą stroną,
- projektowane urządzenia wymaga implementacji stosów USB, Ethernet, Bluetooth,
- zadania stawiane przed naszym urządzeniem, mogą zostać przynajmniej częściowo zrealizowane za pomocą istniejących już pakietów oprogramowania,

Uruchomienie płytki *VisionSOM-6ULL*



Aby uzyskać dostęp do konsoli systemu Linux, należy uruchomić dowolny program emulatora terminalu z ustawieniami transmisji `115200 8N1`:

```
picocom -b 115200 /dev/ttyUSB0
```

Konfiguracja sieci – serwer DHCP

Na potrzeby szkolenia - w stosunku do domyślnej konfiguracji sieci - wprowadzono kilka zmian:

Konfiguracja sieci – serwer DHCP

Na potrzeby szkolenia - w stosunku do domyślnej konfiguracji sieci - wprowadzono kilka zmian:

- zainstalowano pakiet `dnsmasq`, będący lekką implementacją serwera DHCP/DNS:

```
sudo apt-get install dnsmasq
```

Konfiguracja sieci – serwer DHCP

Na potrzeby szkolenia - w stosunku do domyślnej konfiguracji sieci - wprowadzono kilka zmian:

- zainstalowano pakiet `dnsmasq`, będący lekką implementacją serwera DHCP/DNS:

```
sudo apt-get install dnsmasq
```

- poprzez edycję pliku */etc/network/interfaces* do urządzenia przypisano statyczny adres IP:

```
auto eth0
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
```

Konfiguracja sieci – serwer DHCP

Na potrzeby szkolenia - w stosunku do domyślnej konfiguracji sieci - wprowadzono kilka zmian:

- zainstalowano pakiet `dnsmasq`, będący lekką implementacją serwera DHCP/DNS:

```
sudo apt-get install dnsmasq
```

- poprzez edycję pliku */etc/network/interfaces* do urządzenia przypisano statyczny adres IP:

```
auto eth0
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
```

- dokonano modyfikacji pliku konfiguracyjnego */etc/dnsmasq.conf*:

```
interface=eth0
dhcp-range=192.168.0.2,192.168.0.254,255.255.255.0,12h
```