

Platforma I.MX28EVK(2)

Nauka programowania mikrokontrolerów pod Linuksem

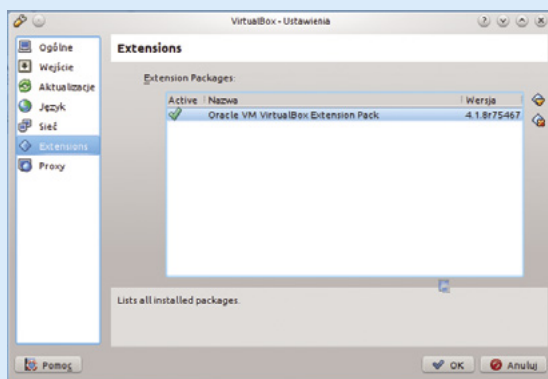
Układy I.MX28 firmy Freescale są jednymi z najmniejszych układów multimedialnych z rodziny I.MX. Układy te wyposażono w rdzeń ARM926EJS (architektura ARMv5TEJ), który jest taktowany sygnałem o częstotliwości 454 MHz. Układ wyposażono w szereg nowoczesnych układów peryferyjnych. W bieżącym odcinku cyklu pokażemy sposób, w który można zainstalować maszynę wirtualną oraz uruchomić najprostszą aplikację.

Do zestawu I.MX28-EVK jest dostarczana płyta DVD z obrazem maszyny wirtualnej opartej o system Ubuntu zawierająca Toolchain służący do przygotowania aplikacji. Dołączenie obrazu maszyny wirtualnej jest bardzo dobrym posunięciem, szczególnie docenią to początkujący użytkownicy. Rozwiązanie to pozwala na zaoszczędzenie czasu potrzebnego na instalację oprogramowania z wykorzystaniem źródła oprogramowania, a użytkownikom systemu Windows umożliwia tworzenie aplikacji bez konieczności instalacji Linuksa na komputerze PC.

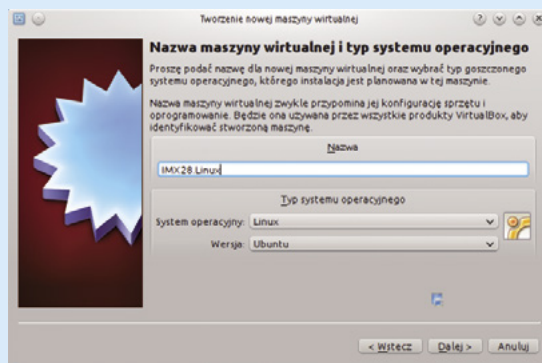
Do pisania aplikacji linuxowych na komputerze PC jest wymagane użycie środowiska Linux, a próby ich tworzenia pod Windows, to prośenie się o nieprzewidywalne i trudne do rozwiązania problemy. Oprogramowanie maszyny wirtualnej stanowi system Ubuntu Linux w wersji 9.04. Toolchain oparty jest o system budowania oprogramowania LTLIB (*Linux Target Image Builder*), który jako system zarządzania oprogramowaniem w systemie docelowym wykorzystuje pakiety RPM (Red-Hat). Trzeba tutaj stwierdzić, że jest to rozwiązanie niszowe, wykorzystywane głównie przez firmę Freescale i stosunkowo rzadko używane na innych platformach. W bieżącym odcinku pokażemy w jaki sposób zainstalować maszynę wirtualną oraz uruchomić najprostszą aplikację.

Instalacja maszyny wirtualnej oraz środowiska

Aby przygotować środowisko dla platformy służącej do tworzenia aplikacji dla zestawu IMX28.EVK, konieczne będzie zainstalowanie maszyny wirtualnej umożliwiającej uruchomienie obrazu systemu. W tym celu spod adresu <https://www.virtualbox.org/wiki/Downloads> należy pobrać



Rysunek 1. Wprowadzenie ścieżki z dodatkami



Rysunek 2. Wybór systemu operacyjnego dla maszyny wirtualnej

aplikację *VirtualBox* dla swojego systemu operacyjnego, a następnie zainstalować ją. Po zainstalowaniu aplikacji, z tej samej strony należy pobrać dodatek *Oracle VM VirtualBox Extension Pack*, następnie uruchomić *VirtualBox* i zainstalować pobrany dodatek. W tym celu należy w menu *File -> Global Settings* wybrać zakładkę *Extensions*, a następnie kliknąć na ikonę ze strzałką oraz podać ścieżkę do pliku zawierającego dodatki (rysunek 1).

Po zainstalowaniu dodatków możemy rozpakować z płyty DVD obraz maszyny wirtualnej, który znajduje się w pliku `/fscommand/Ubuntu_VM_self_extractor.exe`. W systemie Windows obraz możemy rozpakować po prostu klikając na nazwie samorozpakowującego się archiwum. W systemie Linux obraz należy rozpakować za pomocą polecenia `7z x Ubuntu_VM_self_extractor.exe`. Po rozpakowaniu pliku przystępujemy do utworzenia maszyny wirtualnej. W tym celu w menu głównym wybieramy opcję *Maszyna -> Nowa*, co spowoduje uruchomienie kreatora. W oknie kreatora wpisujemy nazwę np. `IMX28.Linux` oraz wybieramy opcję „System Operacyjny: Linux”, „Wersja Ubuntu” (rysunek 2)

Po kliknięciu na przycisk *Dalej* ustalamy wielkość pamięci przydzielonej maszynie wirtualnej. W zależności od pamięci RAM, którą dysponujemy w komputerze ustalamy wielkość w granicach od 512 do 1024 MB. Po wybraniu wielkości pamięci RAM należy wybrać dysk wirtualny maszyny. Aby to zrobić, wybieramy opcję *Użyj istniejącego dysku twardego* oraz podajemy ścieżkę do pliku obrazu, który niedawno rozpakowaliśmy (rysunek 3).

Po przejściu do kolejnego okna będzie wyświetlona informacja zawierająca podsumowanie ustawień, gdzie



Rysunek 3. Wprowadzenie ścieżki do pliku – obrazu

należy wybrać przycisk *Create*, co spowoduje utworzenie nowej konfiguracji. Po przygotowaniu maszyny możemy przystąpić do jej uruchomienia. W tym celu, w oknie głównym klikamy dwukrotnie na ikonę maszyny, co spowoduje jej uruchomienie. Gdy proces uruchamiania zakończy się, zobaczymy ekran logowania. Należy zalogować się podając jako nazwę użytkownika *madfsl* oraz hasło *madfsl*. Ostatnią czynnością przygotowawczą będzie zainstalowanie dodatków, w umożliwiających wygodną zmianę wielkości okna czy kopiowanie/wklejanie pomiędzy hostem a maszyną gospodarzem. W tym celu wybieramy w menu głównym opcję *Urządzenia -> Zainstaluj Dodatki*. Po jej wybraniu w maszynie goszczącej na pulpicie powinna pojawić się ikona płyty CD/DVD, którą należy kliknąć, co spowoduje jej zamontowanie. Następnie trzeba otworzyć program terminalowy (ikona Terminal) i przejść do katalogu z dodatkami wpisując komendę `cd /media/cdrom1`, a następnie wpisać polecenie `sudo ./VboxLinuxAdditions.run`. System poprosi o hasło roota (*madfsl*), a następnie rozpocznie instalację. Instalator może zadawać dodatkowe pytanie, na które możemy pozostawić domyślną odpowiedź.

Po zainstalowaniu dodatków, system jest w zasadzie gotowy do pracy. Aby wprowadzić zmiany należy zrestartować maszynę wirtualną wybierając opcję ponownego uruchomienia komputera. Po uruchomieniu dodatki powinny działać prawidłowo i powinniśmy mieć możliwość zmiany wielkości okna wirtualnego systemu czy kopiowania i wklejania danych pomiędzy aplikacjami systemu gościa i gospodarza.

Kompilowanie przykładowej aplikacji dla maszyny docelowej

Dysponując gotowym środowiskiem, możemy kompilować dowolne aplikacje oraz uruchamiać je na platformie docelowej. Aby zapoznać się ze sposobem pracy, pokażemy jak uruchomić nieskomplikowaną aplikację – zamieszczono ją na **listingu 1**. Jest to typowa aplikacja typu „hello world”, która wypisuje krótkie informację na konsoli tekstowej. Pokażemy teraz w jaki sposób, wspomnianą aplikację uruchomić na platformie IMX28, wykorzystując do kompilacji dostarczone przez producenta środowisko LTLIB.

Pierwszą czynnością, którą należy wykonać jest uruchomienie maszyny wirtualnej, zalogowanie się a następnie kliknięcie w ikonę pulpitu *Terminal in LTLIB directory*. Następnie – korzystając z przeglądarki Mozilla Firefox – spod adresu `ftp://boff.pl/elektronika_praktyczna/imx28/` należy pobrać aplikację `hello-1.0.tar.gz` wraz z plikiem sumy kontrolnej `hello-1.0.tar.gz.md5`. Pobrane pliki należy skopiować do katalogu maszyny wirtualnej `/opt/freescale/pkg/`. Kolejną czynnością jest pobranie pliku `ftp://boff.pl/elektronika_prak-`

Listing 1. Najprostsza, przykładowa aplikacja

```
#include <stdio.h>
int main()
{
    int i;
    printf("hello world\n");
    for ( i = 0; i < 10 ; i++ ) {
        printf("loop count = %d\n", i);
    }
    printf("hello this is the end\n");
    return 0;
}
```

tyczna/imx28/hello.spec, a następnie skopiowanie go do katalogu `~/projects/linux/1008/ltib/dist/lfs-5.1/hello`. Ten plik jest specyficznym plikiem konfiguracyjnym dla systemu LTLIB, opisującym sposób skompilowania oraz przygotowania pakietu. Zawartość pliku zamieszczono na **listingu 2**.

Na początku pliku są zawarte metainformacje, na temat pakietu, nazwy aplikacji, jej autora oraz krótki opis czynności przez nią wykonywanych. Następnie, w kolejnych liniach są zawarte polecenia opisujące poszczególne etapy budowania pakietu. Polecenia zawarte w `%Prep` są wykonywane przed rozpoczęciem kompilowania pakietu i mogą zawierać dodatkowe procedury przygotowawcze, jak na przykład wywołanie skryptu `./configure`. W tym wypadku pole jest puste, ponieważ w tak prostym przykładzie nie musimy wykonywać czynności przygotowawczych. Polecenia zawarte w `%Build` są wykonywane na etapie budowania pakietu, w tym wypadku jest to wywołanie polecenia `make`, które spowoduje skompilowanie aplikacji. Polecenia zawarte w `%Install` zawierają listę czynności, które należy wykonać w celu zainstalowania aplikacji w lokalizacji docelowej. Tu jest to zwykle wywołanie polecenia `cp`, którego zadaniem jest przeniesienie kodu wynikowego w miejsce docelowe. Polecenie `%Clean` jest wywoływane w momencie żądania usunięcia plików pośrednich. W tym wypadku polega ono na usunięciu katalogu, w którym była budowana aplikacja.

Ręczne kompilowanie pakietu z wykorzystaniem systemu LTLIB przebiega w kilku etapach. Pierwszą czynnością, którą należy wykonać jest wywołanie polecenia `./ltlib -m prep -p hello.spec`, którego celem jest rozpakowanie oraz przygotowanie aplikacji do kompilowania. Kolejną czynnością jest przygotowanie pakietu, min. poprzez wywołanie polecenia `%Install` skryptu `spec` oraz przygotowanie docelowego pakietu RPM, co następuje w wyniku wywołania polecenia `./ltlib -m scinstall -p hello.spec`. Ostatnią czynnością jest zainstalowanie pakietu w głównym systemie udostęp-

Listing 2. Zawartość pliku konfiguracyjnego

```
%define pfx /opt/freescale/rootfs/${_target_cpu}
%define __os_install_post %{nil}

Summary      : Hello World test
Name         : hello
Version      : 1.0
Release      : 2
License      : Public Domain, not copyrighted
Vendor       : BoFF
Packager     : Author
Group        : Applications/Test
Source       : %{name}-${version}.tar.gz
BuildRoot    : %{_tmppath}/%{name}
Prefix       : %{pfx}
%Description
%{summary}
%Prep
%setup
%Build
make
%Install
mkdir -p $RPM_BUILD_ROOT/%{pfx}/usr/bin
cp hello $RPM_BUILD_ROOT/%{pfx}/usr/bin/
%Clean
rm -rf $RPM_BUILD_ROOT
%Files
%defattr(-,root,root)
%{pfx}/*
```

Listing 3. Rezultat działania aplikacji

```
hello
hello world
loop count = 0
loop count = 1
loop count = 2
loop count = 3
loop count = 4
loop count = 5
loop count = 6
loop count = 7
loop count = 8
loop count = 9
hello this is the end
```

nionym przez NFS zestawowi EVM28.EVK, co można wykonać za pomocą polecenia `./ltib -m sdeploy -p hello.spec`.

Tak przygotowany pakiet jest już w zasadzie gotowy i w zasadzie możemy przystąpić do uruchomienia go na maszynie docelowej, co można zrobić na kilka sposobów. Na etapie tworzenia oraz uruchamiania oprogramowania najprostszym sposobem będzie udostępnienie całego systemu plików (rootfs) za pomocą serwera NFS, co umożliwi wygodne tworzenie aplikacji bez konieczności ciągłego jej kopiowania pomiędzy maszyną wirtualną a zestawem uruchomieniowym. Innym sposobem jest wykorzystanie do tego celu serwera SSHD, który standardowo jest zainstalowany w obrazie demonstracyjnym. Aby skopiować aplikację Hello World z wykorzystaniem tego sposobu, wystarczy wydać polecenie `scp ./rootfs/usr/bin/hello root@ip_imx28/usr/bin`, w którym `ip_imx28`, to adres IP płytki ewaluacyjnej. Oczywiście, w takim wypadku należy pamiętać o konieczności wcześniejszego skonfigurowania sieci, o czym wspominaliśmy w poprzedniej części. Najprostszym sposobem na skopiowanie aplikacji docelowej jest bezpośrednie skopiowanie jej na kartę SD, do głównego systemu plików. Aby tego dokonać należy do kompu-

tera docelowego podłączyć czytnik kart SD, zamontować partycję systemu plików, a następnie skopiować plik `hello np.` za pomocą polecenia `cp` do lokalizacji docelowej `/usr/bin`. Po skopiowaniu plików należy odmontować kartę pamięci, a następnie umieścić ją ponownie w gnieździe SD zestawu i uruchomić system. Po skopiowaniu pliku możemy sprawdzić działanie aplikacji w systemie docelowym. W tym celu należy zalogować się do konsoli zestawu czy to za pomocą `ssh`, czy konsoli szeregowej, a następnie w terminalu wydać polecenie `hello`, w wyniku którego na terminalu powinien wyświetlić się rezultat działania aplikacji, jak na listingu 3.

Podsumowanie

Wyżej pokazano sposób, w który można skompilować oraz uruchomić przykładową aplikację w zestawie IMX28.EVK wykorzystując dysk maszyny wirtualnej dostarczony przez producenta. Opierając się na tym przykładzie możemy kompilować zarówno aplikację napisane we własnym zakresie, jak i inne projekty przeznaczone dla systemu Linux. System LTLIB sam dba o to, aby wywołać odpowiedni kompilator, a dzięki obrazowi maszyny wirtualnej mamy gotowe środowisko, które w kilka minut możemy wykorzystać w dowolnym systemie operacyjnym. Niestety, oprogramowanie dostarczone przez producenta nieco leciwe, a system LTLIB jest rozwiązaniem niszowym wykorzystywanym głównie przez *Freescale*. W związku z powyższym w wypadku bardziej zaawansowanych prac będziemy zmuszeni do przygotowania odpowiedniego toolchaina samodzielnie, bazując na przykład na środowisku OPENEMEDDED, które zdaniem autora jest dużo lepszym rozwiązaniem niż LTLIB.

Lucjan Bryndza, EP

REKLAMA



Największe targi w branży

AUTOMATICON® 2013

19-22 marca 2013 Warszawa

ZGŁOSZENIA UDZIAŁU DO 30 PAŹDZIERNIKA 2012 R.

XIX Międzynarodowe Targi Automatyki i Pomiarów

BIURO TARGÓW

Al. Jerozolimskie 202, 02-486 Warszawa
tel. 22 874 01 50, 874 02 30
fax 22 874 01 49
e-mail: targi@automaticon.pl

ORGANIZATORZY TARGÓW

PIAP
PRZEMYSŁOWY INSTYTUT
AUTOMATYKI I POMIARÓW

MVM
SP. Z O.O.

www.automaticon.pl

