

Embedded Linux

Радослав Колев <radoslav@kolev.info>

OpenFest 2010

Лиценз: Creative Commons BY-SA 3.0

Какво е embedded система

”Вградена” система е компютърна система конструирана да изпълнява една или няколко специфични функции, често с изискване за работа в реално време. Тя е вградена като част от цяло устройство, обикновено включващо друг хардуер и механични части. В другата крайност са компютрите с общо предназначение, примерно персоналният компютър (РС), проектирани да бъдат гъвкави и да се справят с възможно най-широк кръг нужди.

от Уикипедия

Ограничения

- Размер
- Консумация
- Цена – на разработка и на бройка
- Други – околна среда, надеждност, време за стартиране

Класове системи

- 8/16 битови микроконтролери
- 32/64 битови ARM, MIPS, POWER PC, ...
- x86 – Intel Atom, Via Eria, mini/pico ITX boards
- Комбинация от малък контролер за задачи в реално време + Линукс система за графика, мрежова свързаност и др.

Защо Линукс?

- Гъвкавост
- Солидна основа - стабилно ядро, TCP/IP стек, файлови системи, много драйвери
- Множество библиотеки, компоненти, готов софтуер
- Без лицензни такси
- Достъп до пълния изходен код
- Избор между поддръжка от общността или платена такава

Работи с Линукс, но какво е?



Взето от <http://free-electrons.com/docs/embedded-linux-intro/>

За доене на крави,
доброволно!



Компоненти

- SOC – System on chip – съдържа процесорно ядро, шини за данни, периферия – SPI, I2C, GPIO, UART и много други
- RAM
- FLASH
- Останалият хардуер е специфичен за конкретната система

Хардуер

- развойна платка
- system on module + разширителна платка за конкретния проект
- изцяло нова платка проектирана за конкретния проект – повече работа и трудности - board bring up, не знаеш дали хардуера работи
- ползване на готово устройство– безжичен рутер, NAS

Хардуер

- WiFi Router/NAS
- Biffer board (x86!) - closed source bootloader
- Beagleboard
- Hawkboard
- Digi
- Friendly Arm
- IGEP
- Calao Systems

Софтуер

- Bootstrap и bootloader
- Линукс ядро
- Файлова система

Важно при избора на доставчик на хардуер е каква софтуерна поддръжка е осигурена – BSP – board support package.

Работи ли се по интегриране на промените в "upstream" проектите.

Първоначално зареждане

- 1) BootROM – чете bootstrap код от адрес във flash-а, някои могат и от SD карта
- 2) Bootstrap code - специфичен за SOC-а, инициализира начално хардуера, стартира bootloader
- 3) Bootloader - Uboot, Redboot, Varebox, Linux as the bootloader – зарежда Линукс
- 4) Linux kernel

Крос компилация

- Компилятора произвежда изпълним код за архитектура различна от тази на системата на която се извършва компилирането.
- Съвкупността от всички необходими инструменти се нарича 'toolchain'
- Изграждането на toolchain е трудна задача – много комбинации и версии, зависимости, кръпки. Освен това трябва да се изградят и всички библиотеки.

Помощни инструменти

- Само за toolchain – crosstool-NG
- Цялостни системи за embedded разработка - изграждат toolchain (но може да ползват и външен), ядро, файлова система, приложен софтуер.
- Buildroot, OpenWRT, OpenEmbedded
- PTXdist, ELDK, LTIB, Yocto Project/Poky
- Scratchbox, Aboriginal Linux
- Gentoo embedded, EmDebian, MeeGo

OpenEmbedded

- Bitbake – инструмент на Python, базиран в началото на Gentoo emerge

Метаданни и конфигурация:

- Машини
- Дистрибуции
- Рецепти

Инсталация

- Необходими пакети от дистрибуцията
- `mkdir of_demo; cd of_demo; mkdir -p build/conf`
- `wget http://download.berlios.de/bitbake/bitbake-1.8.18.tar.gz`
- `tar -xzvf bitbake-1.8.18.tar.gz`
- `git clone git://git.openembedded.org/openembedded`
- `dpkg-reconfigure dash` – за Убунту
- `cp openembedded/conf/local.conf.sample build/conf/local.conf`
- `vi build/conf/local.conf`

Инсталация

- `export PATH=/home/rado/tmp/of_demo/bitbake-1.8.18/bin/:$PATH`
- `export BBPATH=/home/rado/tmp/of_demo/build/`
`/home/rado/tmp/of_demo/openembedded/`
- `bitbake console-image`
- Често се прави скрипт, който настройва променливи на обкръжението преди употребата на bitbake. Трябва да се извика със `source` или `.`
- Overlay на цялото дърво за промени свързани с проекта за да се запази основният код чист.

Директорийна структура

- conf – distro, machine, bitbake configuration
- recipes
 - files
 - recipe_name-1.2.3.bb
 - recipe.inc

Директорийна структура

- Работна директория
 - deploy – images, ipk, sdk, sources
 - cross
 - rootfs
 - work
 - recipe_name-1.2.3-r1
 - source code
 - tmp – log files and scripts

Bitbake рецепта

DESCRIPTION = "Hello World Recipe with autotools"

HOMEPAGE = "http://www.example.com"

SECTION = "console/utils"

PRIORITY = "optional"

LICENSE = "GPL"

PR = "r0"

SRC_URI = " \

http://www.mysite.com/downloads/helloworld-\${PV}.tgz \

file://nocomma.patch;patch=1 \

"

inherit autotools

Особенности

- Много динамични промени в OpenEmbedded
- Busybox vs full utils
- OABI/EABI, FPU support
- libc - glibc, uClibc
- MTD – суров флаш
 - USB/SD card != MTD
 - Разпределяне на натоварването, лоши сектори
 - Файлови системи: JFFS2, SquasFS, UbiFS

Връзка с хардуера

- JTAG
- Сериен порт – 3.3/5V → RS-232
- Ethernet – TFTP
- USB

- Много операции, като записване на информация във флаш паметта, се извършват чрез bootloadera

Връзки

- <http://free-electrons.com/docs/>
- <http://elinux.org/>
- Wiki/community на ползваната платка/SOC

Въпроси?

за контакти <radoslav@kolev.info>