Progetti EDALab

Breve introduzione ai progetti di tesi da condurre in EDALab

Novembre 2010

Sommario

Di seguito sono riportate le descrizioni di progetti attivabili a titolo di tesi di primo livello e progetti di laboratorio attinenti ai sistemi embedded e contestualizzati nell'ambito operativo di EDALab.

I progetti riguardano lo sviluppo di applicazioni software per sistemi basati su architettura ARM, seguendo quelle che sono le tendenze di tecnologie emergenti che vedono la disponibilità di hardware a costo sempre più ridotto, librerie software e sistemi operativi evoluti e strumenti di sviluppo complessi. Obiettivo quindi dei progetti descritti di seguito è quello di fare chiarezza sull'efficacia di alcune soluzioni tecnologiche effettuando analisi e comparazioni e arrivando a conclusioni sulla base di implementazioni dimostrative.

Le architetture ARM costituiscono un punto di riferimento per il mondo dei sistemi embedded. L'introduzione della famiglia di processori Cortex, che si propone di coprire tutte le esigenze possibili per lo sviluppo di applicazioni dedicate, ha fatto sì che in tutti i settori della filiera della progettazione e della componentistica periferica vi fosse una risposta in termini di adeguamento e aggiornamento delle soluzioni e del software.

EDALab s.r.l. - Networked Embedded Systems
Cà Vignal, 2 - Strada Le Grazie, 15 - 37134 Verona (VR) – Italy
P.IVA e C.F. 03706250234 - Nr REA: VR 358813 - Cap. Soc. 10000 euro i.v.
Telefono: +39 045 8027062 - Fax: +39 045 8027062 - www.edalab.it - info@edalab.it

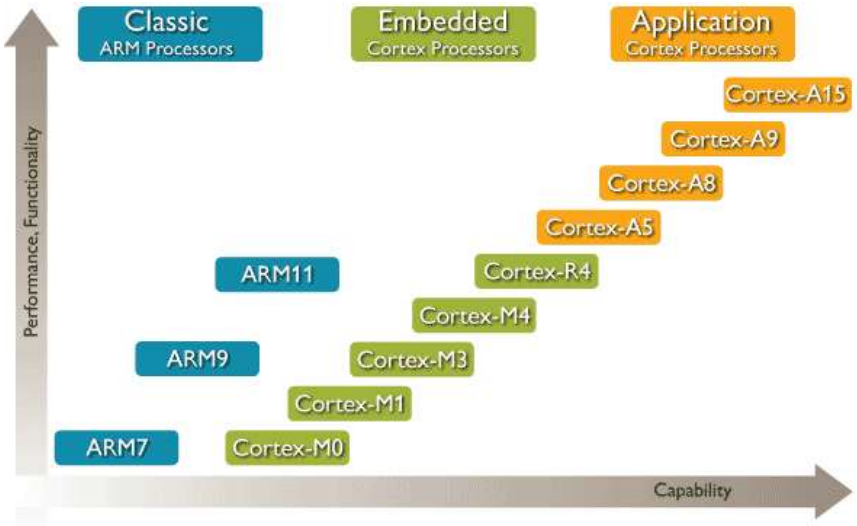

Figura 1 - La nuova famiglia di processori Cortex differenziata per potenza di elaborazione, consumo di potenza e funzionalità accessorie

Sistemi embedded

Sebbene un sistema embedded possa avere forme delle più disparate, è comune considerarlo composto da un insieme di elementi che trovano sicuramente posto in qualsiasi sistema:

- microprocessore o microcontrollore
- storage e memoria di tipi diversi
- interfacce di comunicazione dati
- interfaccia utente (LCD, touchscreen)
- porte di attuazione e sensoristica
- applicazione software con o senza sistema operativo

La scelta di questi componenti è determinata in fase di progettazione sulla base delle specifiche e sulla base degli strumenti necessari alla loro gestione e programmazione, se questi ad esempio sono disponibili al progettista e se non lo sono a quale costo possono essere acquistati. Gli strumenti utili alla progettazione di un sistema embedded, una volta che l'architettura di riferimento è definita sono:

- Ambiente di sviluppo integrato dell'applicazione
  - *toolchain*: linguaggio, compilatore, linker ecc
  - Editor (grafico o testuale)
  - Debugger
- Strumento di trasferimento dei dati sulla memoria del processore
- Capacità di *debugging* supportata dalla scheda di sviluppo.
- Simulatore

La progettazione di un sistema embedded consiste quindi nella capacità di integrare tutte le componenti in modo ottimale ed utilizzando gli strumenti più efficienti. Per poter fare questo occorre avere la conoscenza quanto più approfondita di tutte le parti.

Figura 2 - Esempio di scheda di sviluppo minimale per ARM

Applicazioni Embedded

Lo sviluppo dell'applicazione software è una parte importante in quanto spesso determina il comportamento funzionale del sistema e quindi buona parte della specifica di progettazione. Questo è vero anche per il fatto che è la parte che può variare a costo minore rispetto alle altre componenti e l'impiego di strumenti adeguati e la scelta ragionata di librerie a sistema operativo hanno un grande rilievo in questo. Il linguaggio di programmazione utilizzato per lo sviluppo di applicazioni embedded è prevalentemente è il linguaggio C, soprattutto quando si utilizzano strumenti open source. Alternative sono disponibili con prodotti commerciali o quando sono disponibili macchine virtuali in stile Java Virtual Machine.

Sistemi Operativi

L'impiego di un sistema operativo permette di semplificare enormemente lo sviluppo di applicazioni embedded per i seguenti motivi:

- fornisce delle primitive di sistema di accesso alle risorse hardware che lo sviluppatore dovrebbe fare autonomamente
- permette di rendere l'applicazione indipendente dall'hardware (portabilità su schede diverse)
- semplifica l'aggiornamento del sistema
- permette l'impiego di nuove periferiche tramite driver senza modificare l'applicazione in modo sostanziale

Questi vantaggi derivano principalmente dalla separazione dei contesti e delle relative problematiche che caratterizzano l'applicazione nel suo complesso, ed è il motivo per il quale in Linux, ad esempio, si parla di User Space e Kernel Space.

Gli stessi vantaggi si possono ottenere in contesti più semplici con l'utilizzo di librerie messe a disposizione dai produttori di hardware e hanno la stessa funzionalità del livello di astrazione dello hardware (*HAL:* Hardware Abstraction Level) nei sistemi operativi. Un esempio è la librerie CMSIS (*Cortex Microcontroller Software Interface Standard*) sviluppata e mantenuta da ARM e disponibile gratuitamente per un gran numero di dispositivi basati su architettura Cortex-M. Ogni produttore di schede personalizza alcuni file di configurazione da importare nei progetti e la rende disponibile per il proprio prodotto, consentendo quindi di poter portare applicazioni software che usano la libreria su piattaforme diverse.

Gestione memoria

Il linguaggio C è usato diffusamente per le applicazioni embedded per la possibilità di gestire in modo molto diretto aspetti legati all'uso di memoria. Conoscere quindi le istruzioni del linguaggio e il comportamento del compilatore nel gestirle è fondamentale per un qualsiasi progetto embedded. Esistono tecniche di riduzione dell'occupazione della memoria e strumenti dedicati alla risoluzione di problemi specifici. Esistono versioni diverse del linguaggio che rispondono a criteri di ottimizzazione o adattamento specifici ad architetture hardware particolari.

Real Time

Un aspetto molto interessante dei sistemi embedded è la possibilità di gestire in modo diretto le tempistiche di reazione dell'applicazione agendo sulla sincronizzazione, sulla priorità dei processi e degli *interrupt* (*scheduling*). In alcuni contesti in cui l'applicazione deve reagire in modo certo a stimoli e richieste specifiche è richiesto che l'algoritmo di scheduling dei processi sia modificato, e questa modifica può essere agevolata se il sistema operativo fornisce gli strumenti per farlo.

Progetti

Di seguito sono riportati i possibili progetti da condurre, descritti in termini di attività da svolgere e dei risultati attesi.

Comparazione librerie grafiche per sistemi embedded	• Acquisizione documentazione e definizione di "libreria grafica per sistemi embedded" allo stato dell'arte.
	  ○ Interfacce per LCD
	  ○ Modalità di controllo da parte del Microprocessore
	• Selezione librerie grafiche per l'analisi.
	• Acquisizione strumenti di sviluppo.
	• Analisi comparativa delle librerie in termini di:
	  ○ semplicità d'uso
	  ○ qualità del framework
	  ○ qualità strumenti di sviluppo
	  ○ dimensione footprint
	  ○ possibilità di definire elementi grafici
	  ○ compatibilità piattaforma (processore e LCD)
	• Sviluppo di un'applicazione su una scheda da selezionare per dimostrare l'efficacia delle conclusioni.

Applicazioni e sviluppo driver per sistema operativo CoOS	• Analisi caratteristiche del sistema operativo CoOS • Acquisizione ambiente di sviluppo e selezione di una scheda per poter eseguire applicazioni • Implementazione di applicazioni dimostrative sulla scheda seguendo i tutorial • Acquisizione di una periferica esterna (es. scheda Ardunio, LCD alternativo, ecc) e scrittura di un driver per la periferica. • Sviluppo di un'applicazione che utilizzi il driver implementato

Applicazioni per Java ME in ambito SmartPhone	• Analisi caratteristiche della piattaforma Java ME • Acquisizione ambiente di sviluppo e selezione di una scheda/emulatore per poter eseguire applicazioni • Implementazione di applicazioni dimostrative seguendo i tutorial
Applicazione dimostrativa per il Real Time. Confronto tra eCos e VxWorks	• Analisi caratteristiche del O.S eCos e VxWorks • Acquisizione ambiente di sviluppo (toolchain e IDE) e selezione di una scheda/emulatore per poter eseguire applicazioni • Implementazione di un'applicazione realt time per eseguire il confronto
Sviluppo di Applicazione per Android	• Analisi caratteristiche del O.S Android • Acquisizione ambiente di sviluppo (Eclipse + SDK) e selezione di una scheda/emulatore per poter eseguire applicazioni (e.g., scheda Overo Gumstix) • Implementazione di un'applicazione dimostrativa • Gestione di un'interfaccia di comunicazione (e.g., Ethernet, WiFi)

Realizzazione di un servizio software per funzionalità softphone	• Analisi ed estensione specifiche fornite
	• Acquisizione SDK e installazione ambiente di sviluppo
	• Implementazione di un servizio di sistema che esponga funzionalità softphone per piattaforme a scelta tra Linux, Windows e Linux Embedded con:
	  ○ interfaccia di test
	  ○ gestione con file di configurazione
	  ○ installer e gestione del servizio

Riferimenti

Librerie Grafiche	Librerie Qt	http://qt.nokia.com/products
	Librerie MiniGUI 3.0	http://www.minigui.com
	Prism	http://www.bwembedded.com
	Segger emWin	http://www.segger-us.com/
	EasyGUI	http://www.easygui.com/
	Light Weight User Interface Toolkit (LWUIT)	http://www.oracle.com/technetwork/java/javame/downloads/index.html
Linguaggi di programmazione	Linguaggio C	http://www.acm.uiuc.edu/webmonkeys/book/c_guide/index.html
	Gestione della memoria	http://en.wikibooks.org/wiki/Embedded_Systems/Memory

Sistemi Operativi		
	eCos	http://ecos.sourceware.org
	Java ME	http://www.oracle.com/technetwork/java/javame/overview/index.html
	CoOS	http://www.coocox.com
	Linaro	http://www.linaro.org
	VxWorks	http://www.windriver.com
	HAL - Hardware Abstraction Layer	http://en.wikipedia.org/wiki/Hardware_abstraction_layer
Processori	Atmel	http://www.atmel.com/products/at91
	NXP	http://ics.nxp.com/microcontrollers
Strumenti software	Eclipse	http://www.eclipse.org/dsdp
	LPCXpresso	http://ics.nxp.com/lpcxpresso
	Linux4SAM	http://www.at91.com/linux4sam/bin/view/Linux4SAM
	Keil	http://www.keil.com