



Lezione E5

Sviluppo di un sistema “bare-metal”

Sistemi embedded e real-time

25 ottobre 2012

Marco Cesati

Dipartimento di Ingegneria Civile e Ingegneria Informatica
Università degli Studi di Roma Tor Vergata

[Schema della lezione](#)

[Il progetto](#)

[La scheda TS-7250](#)

[L'ambiente SERT](#)

Di cosa parliamo in questa lezione?



In questa lezione cominciamo a sviluppare un piccolo progetto per realizzare un sistema “bare-metal” su una scheda embedded

- 1 Descrizione del progetto
- 2 La scheda TS-7250
- 3 L'architettura ARM
- 4 L'ambiente di esecuzione SERT

Schema della lezione

[Il progetto](#)

[La scheda TS-7250](#)

[L'ambiente SERT](#)



Lo scopo del progetto consiste nella scrittura di un sistema software real-time “bare-metal”:

- Scrittura di un ambiente di esecuzione
 - minimale e auto-contenuto (senza sistema operativo)
 - in grado di schedulare job con requisiti real-time
 - su un hardware “embedded”
- Verifica e validazione del sistema per requisiti hard real-time

[Schema della lezione](#)

[Il progetto](#)

[La scheda TS-7250](#)

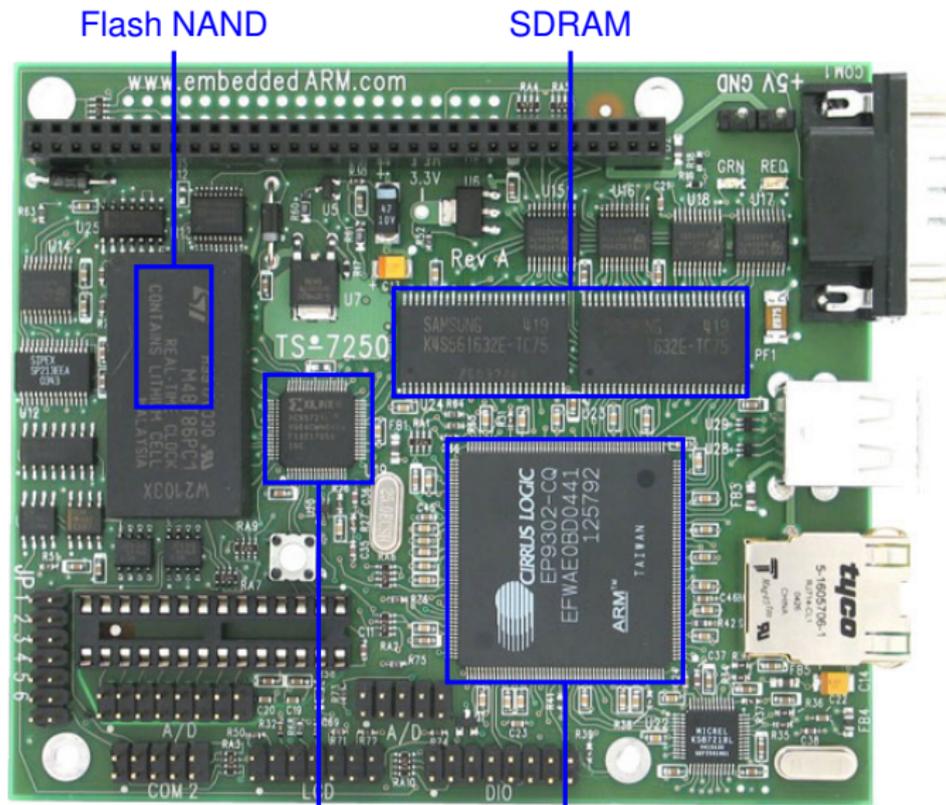
[L'ambiente SERT](#)

La scheda Technologic Systems TS-7250

- È una scheda **SBC** (Single Board Computer)
- Basata su chip Cirrus Logic EP9302 con CPU ARM9
- 32 MB memoria flash
- 32 MB o 64 MB SDRAM
- 2 KB EEPROM
- 2 porte USB 2.0
- 2 porte seriali
- 1 porta Ethernet 100 Mbit/s
- 19 ingressi/uscite digitali
- Convertitore 5 canali A/D a 12 bit
- Scheda di espansione PC-104
- Timer watchdog
- Interfacce: SPI, matrix keypad, LCD
- Temperature di esercizio (senza ventola): da -40° a $+70^{\circ}$



Dispositivi della scheda TS-7250



Flash NAND

SDRAM

CPLD Xilinx

CPU ARM9
e dispositivi di I/O

Sviluppo di un
sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

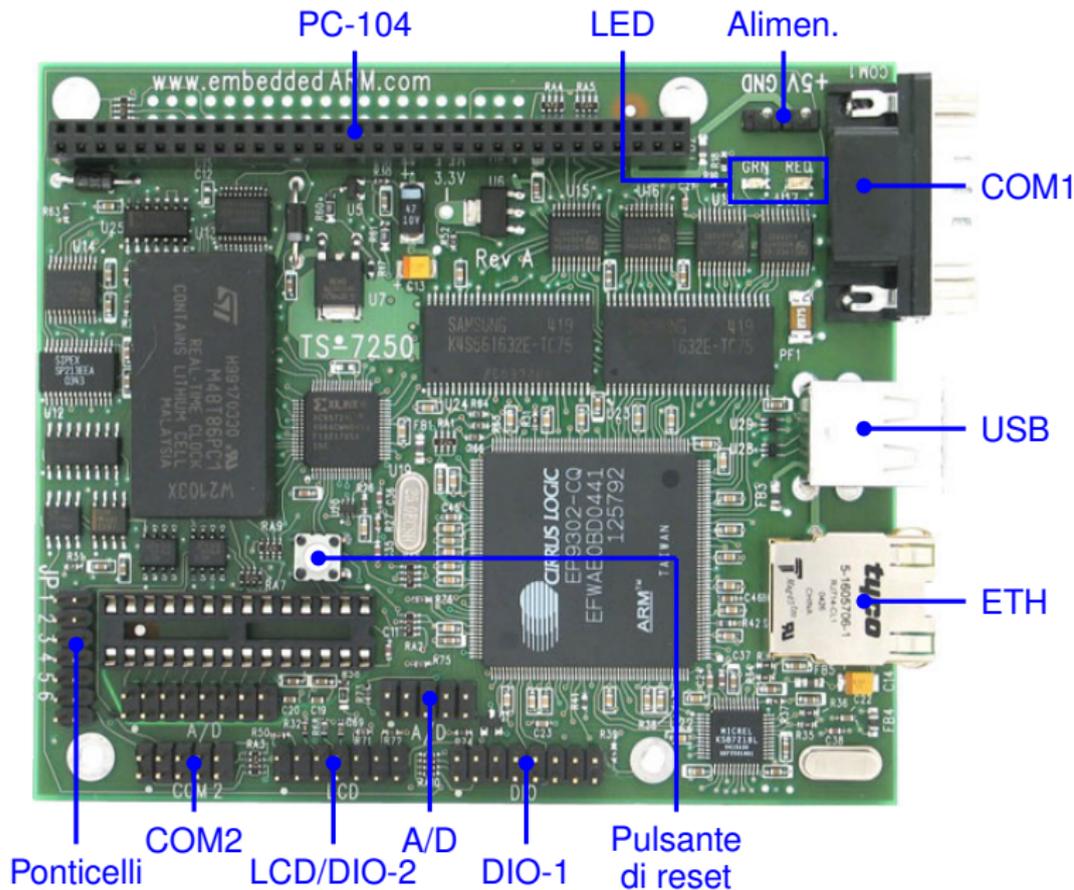
La scheda TS-7250

L'ambiente SERT

Interfacce della scheda TS-7250

Sviluppo di un sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda TS-7250

L'ambiente SERT

Procedura di boot della scheda TS-7250

L'inizializzazione della scheda TS-7250 avviene in cinque fasi:

- Subito dopo l'accensione od il reset della scheda viene eseguito un programma in una ROM nel chip EP9302
- Viene caricato un altro programma chiamato **TS-NANDBOOT** contenuto nella EEPROM:

```
>> TS-NANDBOOT, Rev 1.00 - built 10:14:31, Mar 7 2005  
>> Copyright (C) 2004, Technologic Systems
```

- **TS-NANDBOOT** esegue un programma proprietario chiamato **TS-BOOTROM** memorizzato in flash:

```
>> TS-BOOTROM, Rev 1.05 - built 09:44:15, Mar 8 2005  
>> Copyright (C) 2005, Technologic Systems  
>> TS-7250: Rev E0 CPU, Rev B PLD, 64MB SDRAM
```

- **TS-BOOTROM** esegue un boot loader open-source chiamato **RedBoot** memorizzato in flash:

```
+Ethernet eth0: MAC address 00:d0:69:40:11:9a  
IP: 192.168.1.2/255.255.255.0, Gateway: 192.168.1.1  
Default server: 192.168.1.1  
RedBoot(tm) bootstrap and debug environment [ROMRAM]  
Non-certified release, version current-TS_5 - built 16:49:40,...  
Platform: TS-7250 Board (ARM920T) Rev A  
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.
```



Schema della lezione

Il progetto

La scheda TS-7250

L'ambiente SERT

Procedura di boot della scheda TS-7250 (2)

La quinta fase del bootstrap dipende da **RedBoot**, che può:

- ricevere comandi dall'utente sulla linea seriale COM1
- automaticamente eseguire una sequenza di comandi che carica l'ambiente di esecuzione (sistema operativo) da memoria flash, rete, o porta seriale

```
RAM: 0x00000000-0x02000000, [0x00051570-0x01ff9000] available
FLASH: 0x60000000 - 0x68000000, 8192 blocks of 0x00004000 bytes each.
== Executing boot script in 0.300 seconds - enter ^C to abort
RedBoot> fis load vmlinux
RedBoot> exec
Using base address 0x00218000 and length 0x000a5f7c
Linux version 2.4.26-ts9 (root@ronald-pc) (gcc version 3.3.4) #1 ...
CPU: Arm920Tid(wb) revision 0
Machine: ep9301
:
:
Technologic Systems TS-LINUX/arm 7.0
ts7200 login:
```



[Schema della lezione](#)

[Il progetto](#)

[La scheda TS-7250](#)

[L'ambiente SERT](#)

Il processore della scheda TS-7250

Il chip [Cirrus Logic EP9302](#) include una CPU [ARM920T](#) con:

- Architettura RISC a 32 bit
- Set di istruzioni ARM (32 bit) e Thumb (16 bit)
- Frequenza di lavoro a 200 MHz
- Pipeline delle istruzioni a 5 stadi
- Cache separata per le istruzioni da 16 KB
- Cache separata per i dati da 16 KB
- Memory Management Unit (MMU) per il supporto alla memoria virtuale
- Translation Look Aside (TLA) Buffers con 64+64 voci
- Pagine di memoria da 1 MB, 64 KB, 4 KB e 1 KB



[Schema della lezione](#)

[Il progetto](#)

[La scheda TS-7250](#)

[L'ambiente SERT](#)



SERT (System Environment for Real-Time applications)

è (sarà) un ambiente di esecuzione per la scheda TS-7250:

- “Bare-metal”: si interfaccia direttamente all’hardware
- Compatto: solo le funzionalità strettamente necessarie
- Real-time: con prestazioni predicibili, verificabili e validabili
- Semplice: la maggior parte del codice è
 - in linguaggio C
 - scritto e commentato “in diretta”

[Schema della lezione](#)

[Il progetto](#)

[La scheda TS-7250](#)

[L'ambiente SERT](#)

Sviluppo di codice per sistemi embedded



- Lo sviluppo di codice di un sistema embedded è svolto su di un calcolatore general-purpose (sistema *host*)
 - È necessario installare un ambiente di sviluppo (*cross-compiler*) in grado di generare codice macchina per il sistema embedded
- I programmi si interfacciano direttamente con l'hardware
 - È conveniente utilizzare linguaggi a basso livello come l'Assembler e il "C"
 - È necessario avere a disposizione la documentazione completa del processore e di tutte le periferiche del sistema utilizzate
- Generalmente la scheda embedded non ha interfacce di comunicazione "amichevoli"
 - È essenziale ottenere feedback dal codice per effettuare il debug e le verifiche di funzionamento
 - Inizialmente il miglior canale di feedback è un led luminoso
 - Appena possibile si dovrebbe utilizzare un canale come la porta seriale per ottenere messaggi di stato dal sistema