

11/10/2022

# Lezione E3a

## Sviluppo di un sistema "bare-metal"

Sistemi embedded e real-time

27 novembre 2020

Marco Cesati

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

### Di cosa parliamo in questa lezione?

Vogliamo definire un **sistema sw**, cioè un ambiente di esecuzione (e NON s.o) **real time Bare Metal**

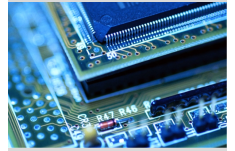
In questa lezione cominciamo a sviluppare un piccolo progetto per realizzare un sistema "bare-metal" su una scheda embedded

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

esso è ambiente MINIMALE e **AUTO CONTENUTO** (durante il Bootstrap, non avendo il s.o, non esiste nulla: funzioni libreria, di sistema...) L'ambiente di esecuzione dovrà essere validato come HRT.

Sviluppo di un  
sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone Black

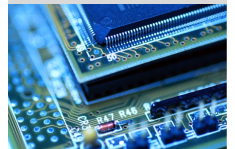
L'ambiente SERT

SERT'20

E3a.1

Sviluppo di un  
sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

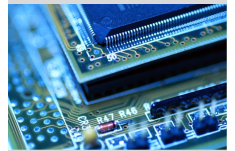
La scheda Beaglebone Black

L'ambiente SERT

SERT'20

E3a.2

## Scopo del progetto



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

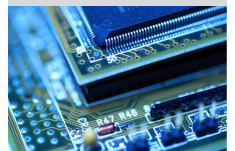
- Scrittura di un ambiente di esecuzione
  - minimale e auto-contenuto (senza sistema operativo)
  - in grado di schedulare job con requisiti real-time
  - su un hardware "embedded"
- Verifica e validazione del sistema per requisiti hard real-time

hardware in cui  
eseguiremo il nostro task

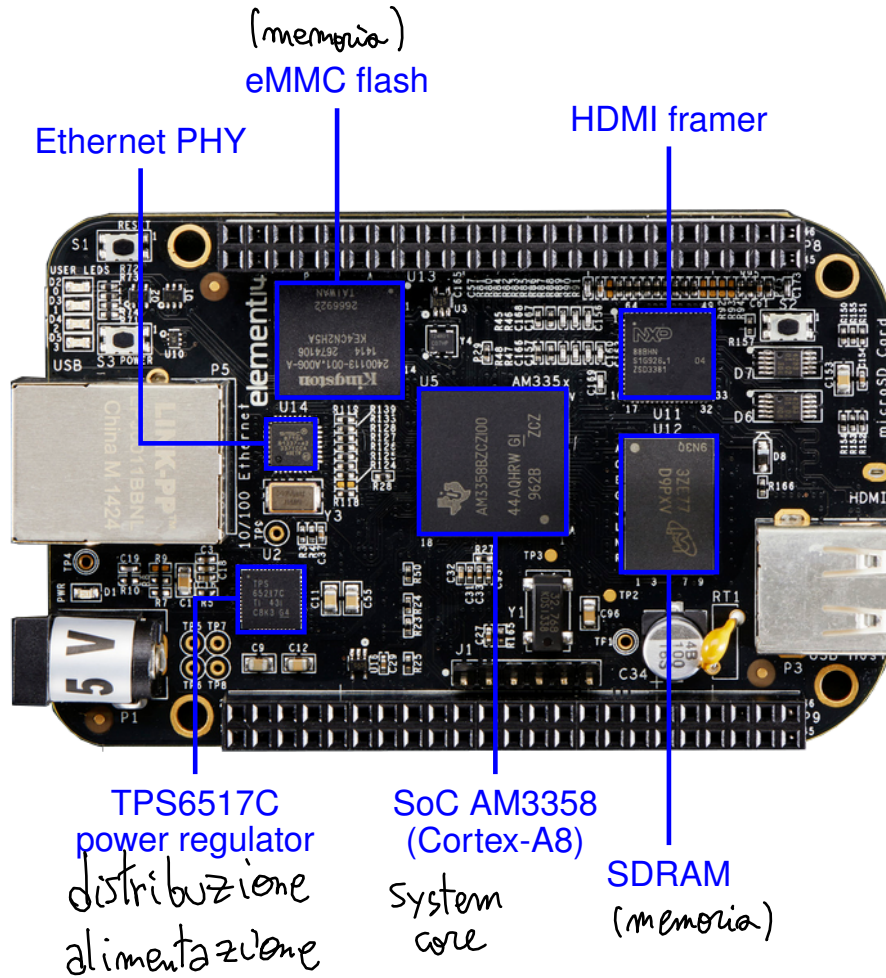
## La scheda Beaglebone Black

- È una scheda **SBC** (Single Board Computer) *simile o  
programmare  
un computer*
- Basata su SoC Texas Instruments AM335x *(senza S.O.)*
- CPU ARM Cortex-A8 @ 1 GHz
- 512 MB DDR3 SDRAM *(per sist. embedded va bene!)*
- 4 GB eMMC on-board flash storage (rev.C)
- acceleratore grafico 3D
- acceleratori virgola mobile VFP, NEON
- 2 PRU (core ARM semplificato a 200 MHz)
- 1 porta USB "client" (alimentazione e comunicazione), in alto a sx.
- 1 porta USB "host", in basso a dx.
- 1 porta Ethernet 100 Mbit/s
- 1 uscita video HDMI, 1280x1024
- 1 porta seriale (3.3V TTL)
- peso: meno di 40 grammi

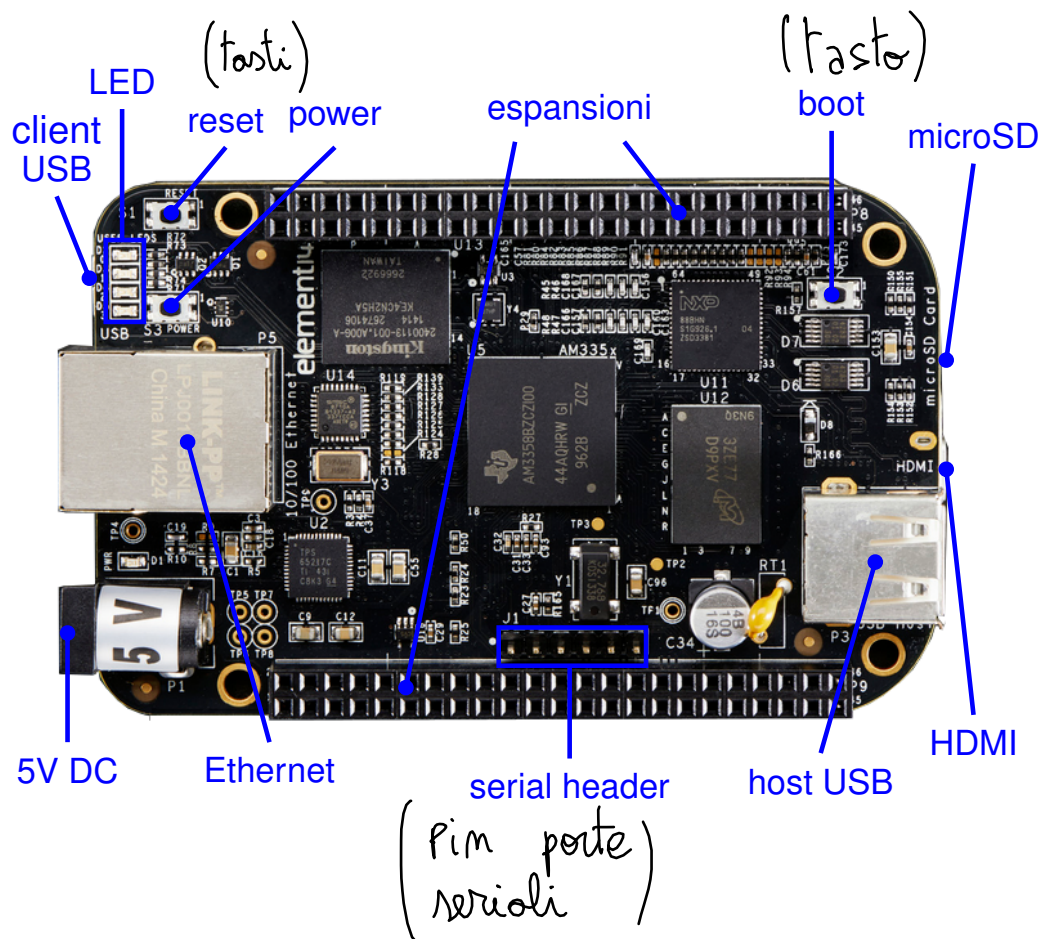
Questa scheda è "open source" e  
diffusa: leggibile, modificabile e  
personalizzabile, per adattarla  
al nostro scopo.



## Dispositivi della Beaglebone Black

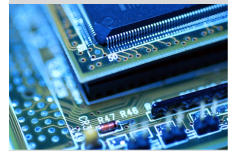


## Interfacce della Beaglebone Black



Sviluppo di un sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone Black

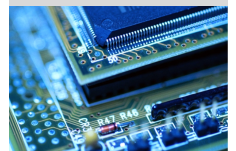
L'ambiente SERT

SERT'20

E3a.5

Sviluppo di un sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone Black

L'ambiente SERT

► No ventole  
ARM scaldano poco,  
a differenza di  
Intel.

SERT'20

E3a.6

## Procedura di boot della Beaglebone Black

L'inizializzazione avviene in diverse fasi: **FASE 1**

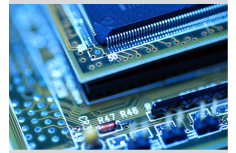
- Subito dopo l'accensione od il reset della scheda viene eseguito un programma in una ROM nel chip AM335x
- A seconda dello stato dello switch "boot", viene caricato un programma dalla flash eMMC oppure da una scheda microSD esterna
- Il programma nella flash eMMC è per default **uBoot**:

```
U-Boot SPL 2019.04-00002-gbb4af0f50f (Jul 08 2019-11:44:39 -0500)
Trying to boot from MMC2
Loading Environment from EXT4...
U-Boot 2019.04-00002-gbb4af0f50f (Jul 08 2019 - 11:44:39 -0500),
Build: jenkins-github_Bootloader-Builder-128
CPU : AM335X-GP rev 2.1
I2C: ready
DRAM: 512 MiB
Reset Source: Global external warm reset has occurred.
Reset Source: Power-on reset has occurred.
RTC 32KCLK Source: External.
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
Loading Environment from EXT4...
Board: BeagleBone Black
BeagleBone Black:
Press SPACE to abort autoboot in 2 seconds
```

open source,  
uBoot carica cio'  
che crede essere il  
sistema bare metal.

Sviluppo di un  
sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone  
Black

L'ambiente SERT

SERT'20

E3a.7

## Procedura di boot della Beaglebone Black (2) FASE 2

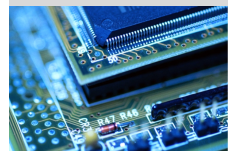
La fase successiva del bootstrap dipende da **uBoot**, che può:

- ricevere comandi dall'utente sulla porta seriale
- automaticamente eseguire una sequenza di comandi che carica l'ambiente di esecuzione (sistema operativo) da
  - memoria flash
  - rete
  - porta seriale
  - USB
  - microSD
- Nel mondo ARM non viene utilizzato un firmware che interroga le periferiche hardware e fornisce una lista al SO
- Si utilizza invece una descrizione statica delle periferiche esistenti sulla scheda
  - In Linux: **Device Tree**

il Bios è semplificato,  
è solo un elenco statico  
di periferiche presenti nella scheda!

Sviluppo di un  
sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone  
Black

L'ambiente SERT

SERT'20

E3a.8



## Esempio di avvio di Linux da flash eMMC

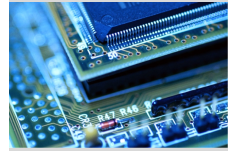
```
board_name=[A335BNLT] ...
board_rev=[00C0] ...
SD/MMC found on device 0
Checking for: /uEnv.txt ...
205 bytes read in 10 ms (19.5 KiB/s)
Loaded environment from /uEnv.txt
Importing environment from mmc ...
Checking if uenvcmd is set ...
Running uenvcmd ...
2708952 bytes read in 232 ms (11.1 MiB/s)
35786 bytes read in 18 ms (1.9 MiB/s)
## Booting kernel from Legacy Image at 82000000 ...
Image Name: Linux-4.14.74
Created: 2018-10-14 12:47:31 UTC
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 2708888 Bytes = 2.6 MiB
Load Address: 82000000
Entry Point: 82000000
Verifying Checksum ... OK
## Flattened Device Tree blob at 88000000
Booting using the fdt blob at 0x88000000
Loading Kernel Image ... OK
Loading Device Tree to 8fff4000, end 8ffffbc9 ... OK
Starting kernel ...

:
:

Welcome to Linux 4.14.74 armv7l (ttyS0)
BONEBLACK login:
```

Sviluppo di un  
sistema bare-metal

Marco Cesati



[Schema della lezione](#)

[Il progetto](#)

[La scheda Beaglebone Black](#)

[L'ambiente SERT](#)

SERT'20

E3a.9

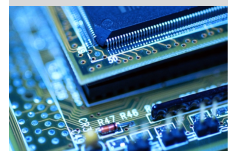
## Il processore della Beaglebone Black

Il chip **TI AM3358** include una CPU **ARM Sitara Cortex-A8** con:

- Architettura RISC a 32 bit
- Set di istruzioni ARM32, Thumb2, Jazelle
- Frequenza di lavoro a 1 GHz
- Cache integrata L1 32+32 KiB (istruzioni e dati)
- Cache integrata L2 da 256 KiB
- Memory Management Unit (MMU) per il supporto alla memoria virtuale
- Translation Lookaside Buffers (TLB) con 32+32 voci
- Branch target address cache con 512 voci (predice il target delle istruzioni di tipo salto condizionale o indiretto)
- Coprocessore VFP (floating point)
- Coprocessore NEON Media (istruzioni SIMD)
- Pagine di memoria da 16 MiB, 1 MiB, 64 KiB, 4 KiB

Sviluppo di un  
sistema bare-metal

Marco Cesati



[Schema della lezione](#)

[Il progetto](#)

[La scheda Beaglebone Black](#)

[L'ambiente SERT](#)

SERT'20

E3a.10

## L'ambiente di esecuzione SERT

SERT (System Environment for Real-Time applications)

è un ambiente di esecuzione già funzionante per

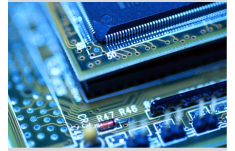
- Technologic Systems TS-7250
- Raspberry PI
- Beaglebone Black

Caratteristiche:

- "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 sarà:
  - in linguaggio C & in Assembler, poiché interfacciarsi con HW richiede linguaggi a basso livello.
  - scritto e commentato "in diretta"

Sviluppo di un sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone Black

L'ambiente SERT

SERT'20

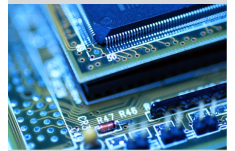
E3a.11

## Sviluppo di codice per sistemi embedded

- Lo sviluppo di codice di un sistema embedded è svolto su di un calcolatore general-purpose (sistema *host*), *non diretto sull'HW*
  - 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" (no schermo, tastiera.... proprio nulla!)
  - 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

Sviluppo di un sistema bare-metal

Marco Cesati



Schema della lezione

Il progetto

La scheda Beaglebone Black

L'ambiente SERT

SERT'20

E3a.12