

Corso di Architettura dei Sistemi a Microprocessore

Introduzione al Corso



Luigi Coppolino

Contact info

Prof. Luigi Coppolino
luigi.coppolino@uniparthenope.it

Università degli Studi di Napoli "Parthenope"
Dipartimento di Ingegneria

Centro Direzionale di Napoli, Isola C4
V Piano lato SUD - Stanza n. 512

Tel: +39-081-5476702
Fax: +39-081-5476777

Roadmap

- Supporti didattici
- Metodologia didattica
- Modalità d'esame
- Tentative Schedule
- Obiettivi del corso
- Contesto di riferimento
- Acknowledgements
- Domande

Supporti didattici

- Libro di testo:
 - Hamacher, Vranesic, Zaky – “Computer Organization”, (6th edition), McGraw-Hill (eng)
- Testo suggerito:
 - Hamacher, Vranesic, Zaky – “Introduzione all'architettura dei calcolatori “
(3° edizione), Mcgraw-Hill (ita)
- Ambienti didattici di supporto:
 - Easy68K
 - ARMSim
 - Logisim (per esercitarsi sulle reti logiche)
 - ChibiOS Studio (per sviluppo su board STM32F4/3 discovery)



Metodologia didattica: Blended Teaching

- Lezioni frontali con supporto multimediale
 - Esercitazioni guidate in aula con due architetture di riferimento: M68000, ARM7
 - Esercitazioni assistite in laboratorio
 - Esercitazioni su una board reale basata su architettura ARM: STM32F4-Discovery
- Lezioni eLearning
 - Approfondimenti alle lezioni teoriche e esercitazioni guidate

Formato delle dispense

- Le dispense saranno rese disponibili in uno dei seguenti formati:
 - Postscript
 - Estensione: ps
 - Visualizzatore: ghostview
 - Portable Data Format
 - Estensione: pdf
 - Visualizzatore: Acrobat Reader
 - PowerPoint
 - Estensione: ppt
 - Visualizzatore: Microsoft PowerPoint

Sito web del corso

<http://edi.uniparthenope.it/>

- **Corso: Architettura dei Sistemi a Microprocessore**
 - Iscriverti al Gruppo ASM 2015/2016 (*proviamo insieme*)
- Il materiale del corso frontale si trova nella sezione CORSO FRONTALE, cartella SLIDE
- Il material del corso online è accessibile direttamente dalla home del corso

Modalità d'esame (tentativo)

- Elaborato basato sulla board STM32F4-Discovery
 - Fornita in aula a ciascuno studente durante le esercitazioni
- Prova al calcolatore (intercorso)
 - Esercizio su architetture 68k e ARM mediante simulatore
- Discussione



Schedule

➤ Orario delle lezioni:

Lunedì 9.00-11.00 Aula inf-4

Mercoledì 11.00-13.00 Aula inf-4

Giovedì 13.00-16.00 Aula inf-3

➤ Organizzazione delle lezioni:

Inizio 29/02/2016

Fine 31/05/2016

Obiettivi del corso

- Fornire le conoscenze di base necessarie relative a:
 - Architettura dei calcolatori elettronici
Quali sono i componenti di un calcolatore, come sono interconnessi, come interagiscono per portare a termine l'elaborazione
 - Linguaggio del processore
Le istruzioni del processore, la programmazione in linguaggi assemblativi, corrispondenza tra linguaggi ad alto livello e linguaggio macchina
 - Sviluppare sistemi hw/sw mediante l'ausilio di microcontrollori
- Non quello di creare programmatori assembler esperti



USO DEI microprocessori

In quali applicazioni?



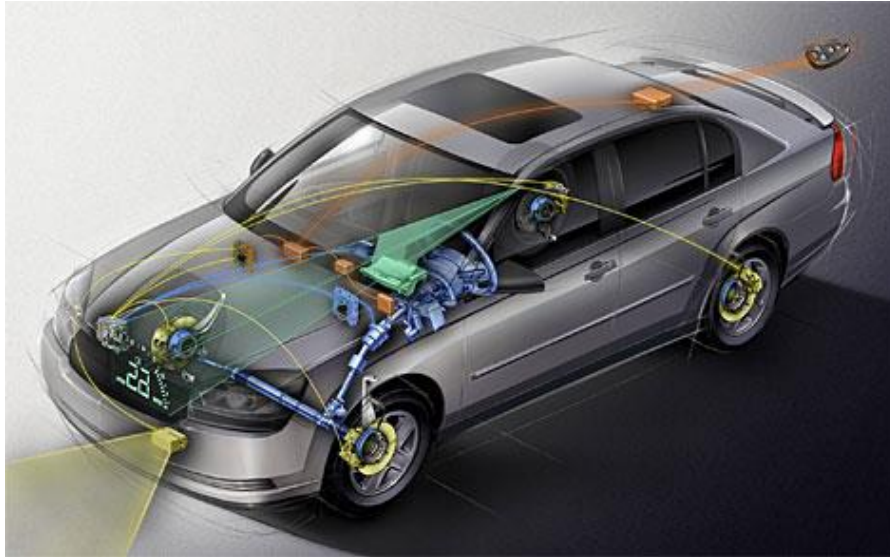
The Fault and Intrusion Tolerant NEtworked SystemS (FITNESS) Research Group

<http://www.fitnesslab.eu/>





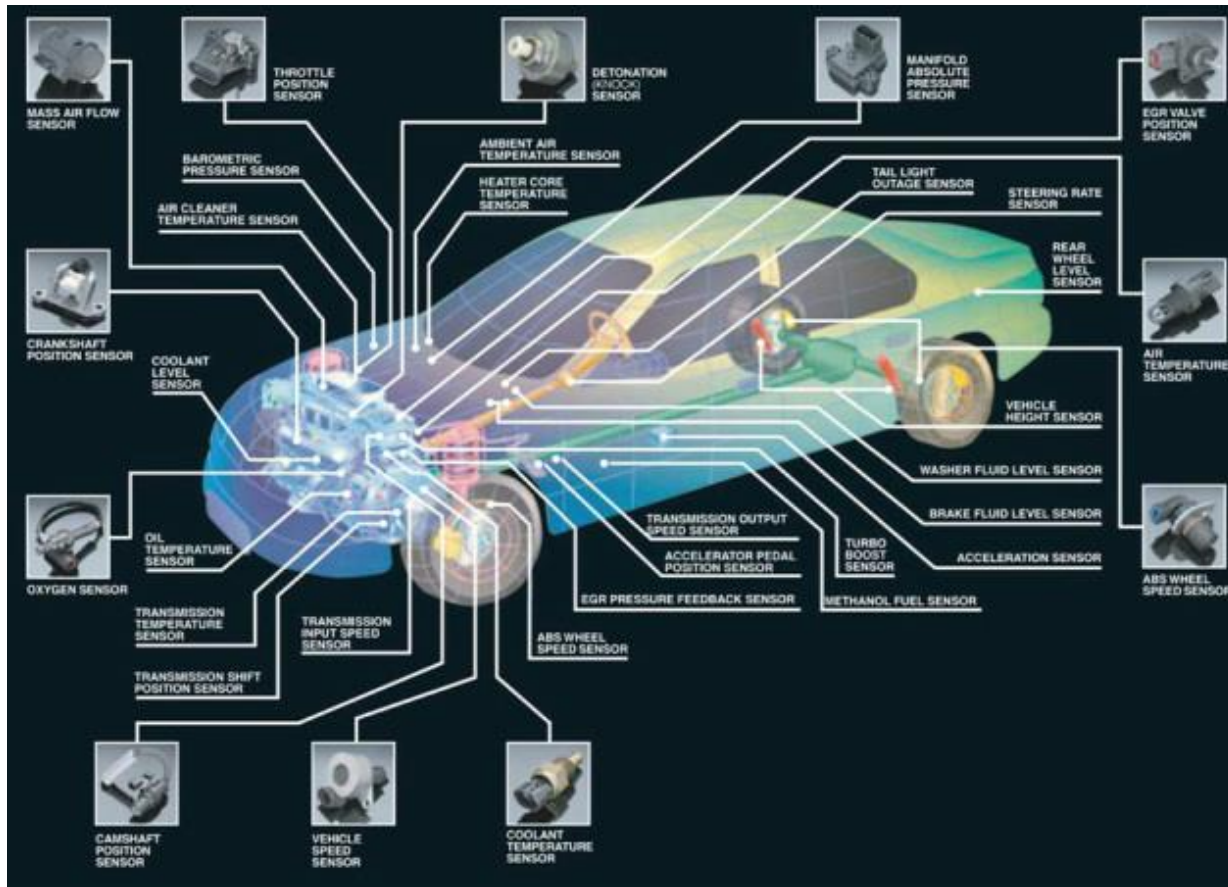
<http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code>



F-22 Raptor, the current U.S. Air Force frontline jet fighter, consists of about 1.7 million lines of software code

Boeing's 787 Dreamliner requires about 6.5 million lines of software code to operate its avionics and onboard support systems.

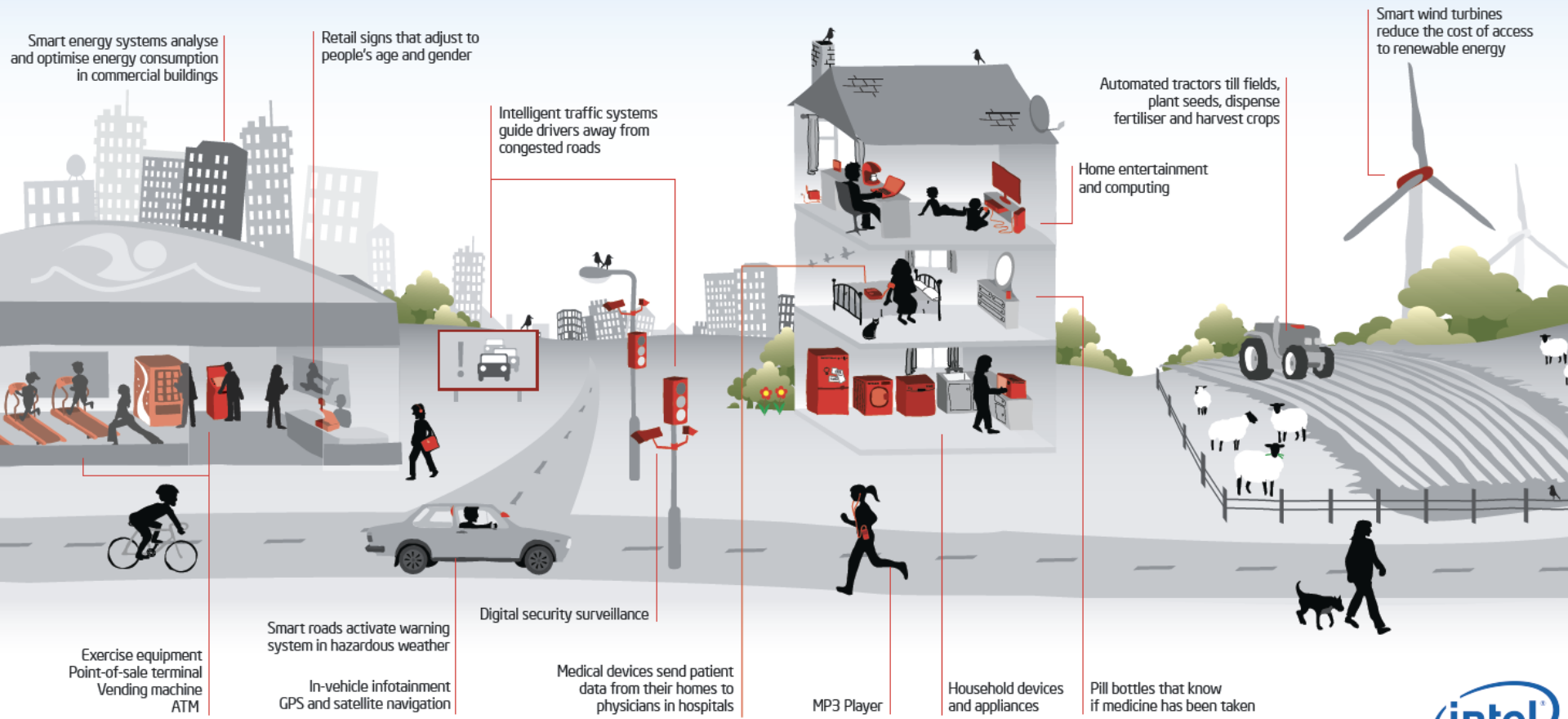
A premium-class automobile recently, contains close to 100 million lines of software code, that executes on 70 to 100 microprocessor-based electronic control units (ECUs) networked throughout the body of your car.



low-end cars now have 30 to 50 ECUs embedded in the body, doors, dash, roof, trunk, seats,

It's a Smart World

Invisible yet ubiquitous, small but mighty, unnoticed but life changing. Forty years ago the microprocessor was born, beginning the quiet but profound process which has radically reshaped our lives. Today, thanks to the microprocessor, we live in a smart world, can do smart things and make smart choices. We don't see them, but these tiny embedded computers shape our world to a remarkable degree. From the cars we drive and tractors that till the fields, to the fresh food delivered to our shops, billboards that advertise and machines that help us stay fit - they're the invisible brains that power our daily being. Long live the smart life.

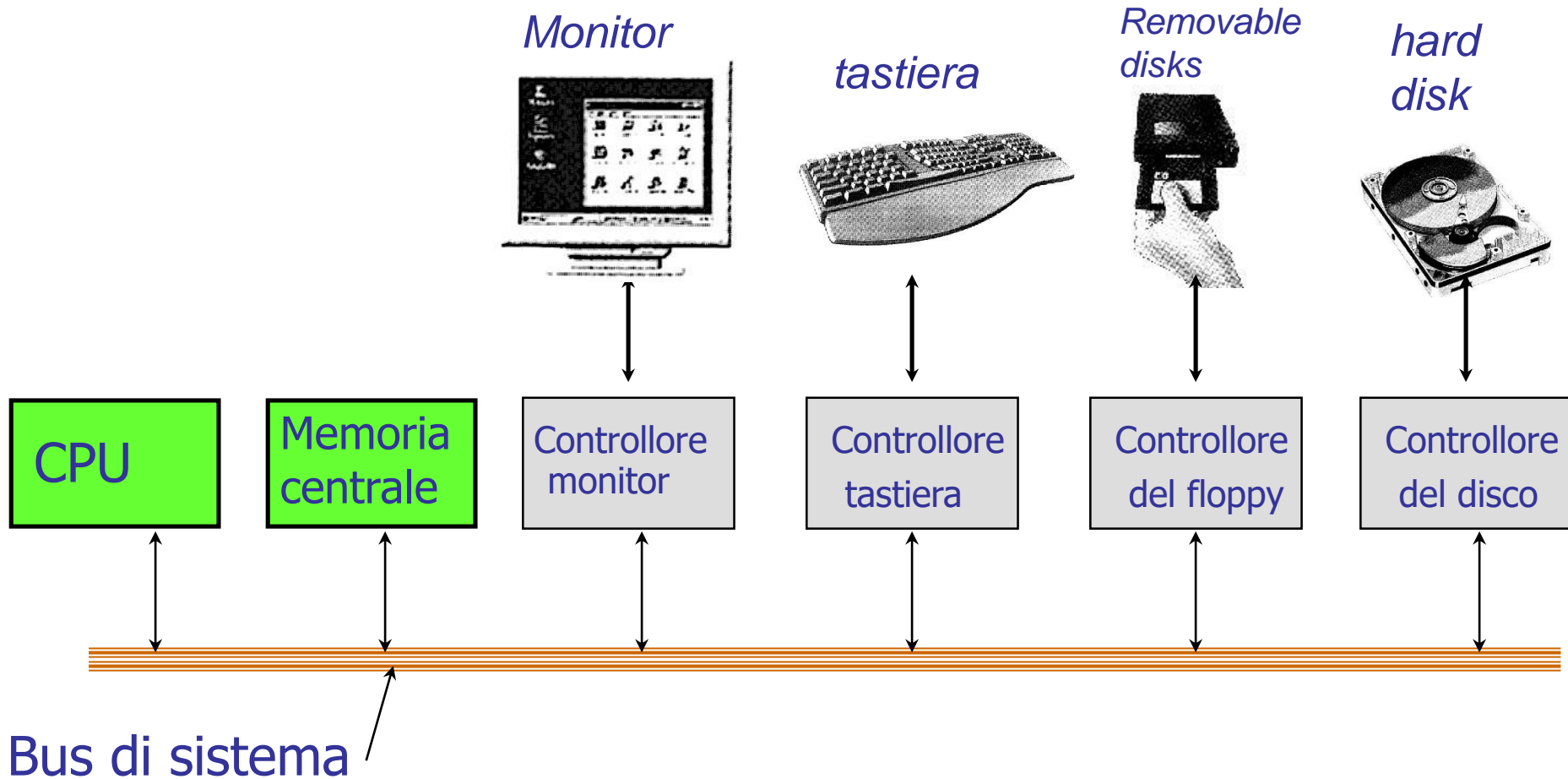
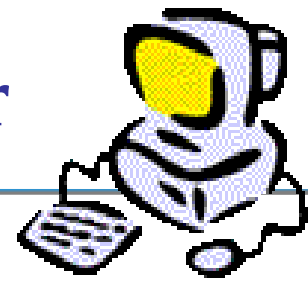


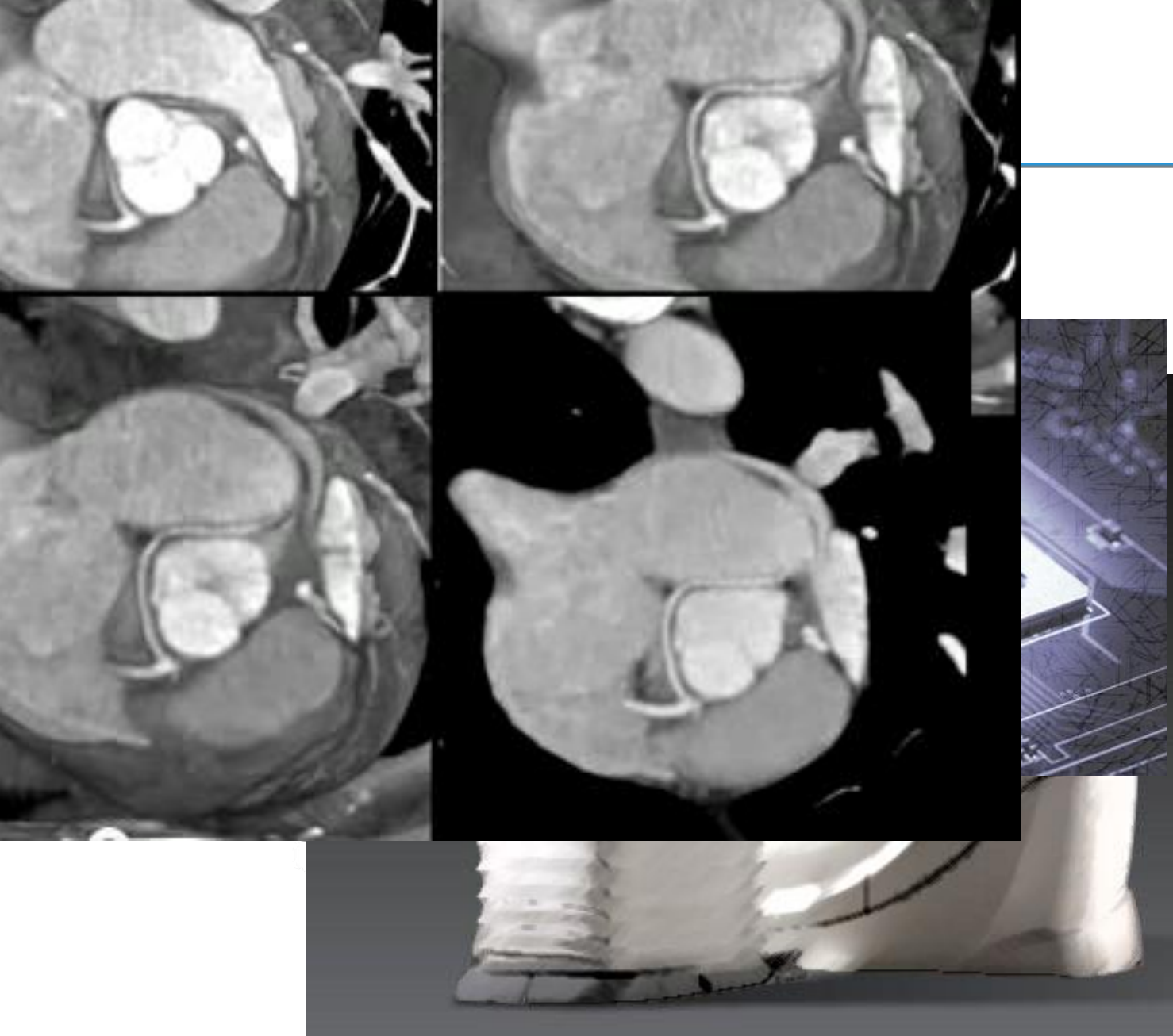
The Fault and Intrusion Tolerant NETworked SystemS (FITNESS) Research Group

<http://www.fitnesslab.eu/>



Componenti di un Personal Computer







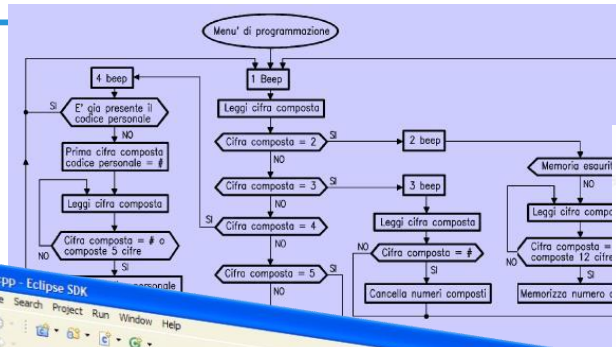
Come si può definire un sistema a microprocessore

Esecutore di Applicazioni



Interprete di Programmi

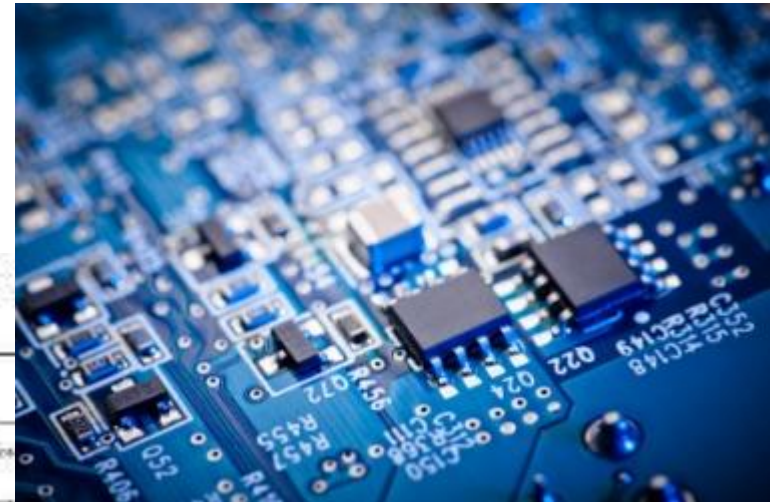
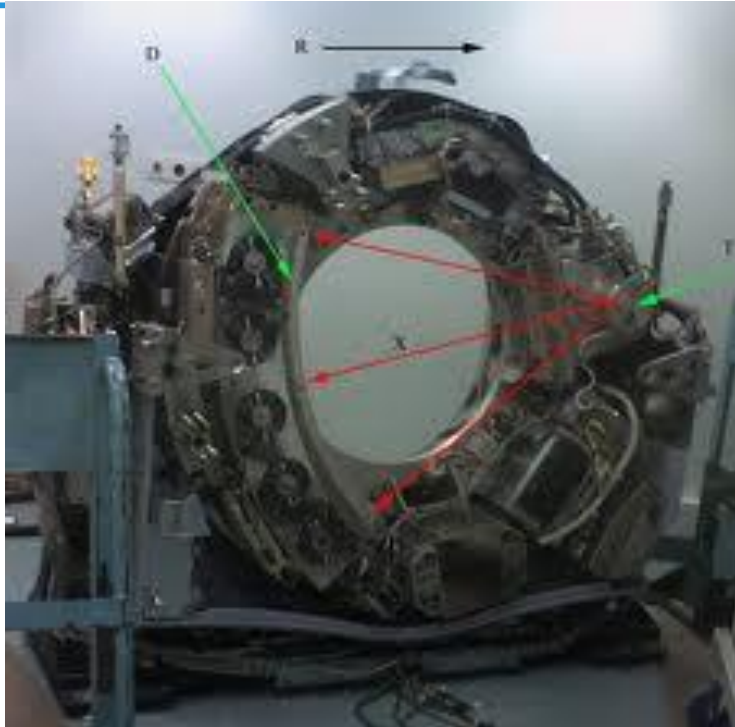
The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Refactor, Navigate, Search, Project, Run, Window, and Help. The left-hand side contains a Project Explorer showing a C/C++ project with folders like ConfigFiles, USB_Device, USB_Driver, and USB_FRPGA. The main editor window displays the source code for `usbp_cyusb.cpp`, showing a `controlWrite` function. The console at the bottom shows a warning: "Error launching external scanner info generato USBP_LIB".



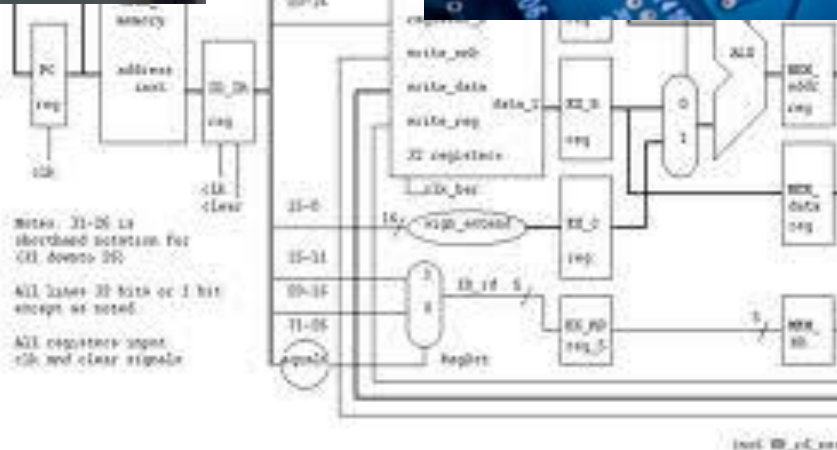
```
<!DOCTYPE HTML>
<html>
<head>
<meta name="title">
<meta name="KEYWORDS">
<meta name="DESCRIPTION">
<link rel="stylesheet" type="text/css" href="style.css">
<script language="JavaScript" src="script.js" color="#ffff">
</head>
```

The terminal window displays a large volume of system logs, including kernel messages, network interface status, and system boot information. The text is dense and appears to be a raw log output from a system.

Un Sistema Hardware



Project Part1



The Fault and I

<http://www.fitnesslab.eu/>

Research Group



Macchina a strati

User Level: Application Programs

Problem-oriented Languages

Assembly Language

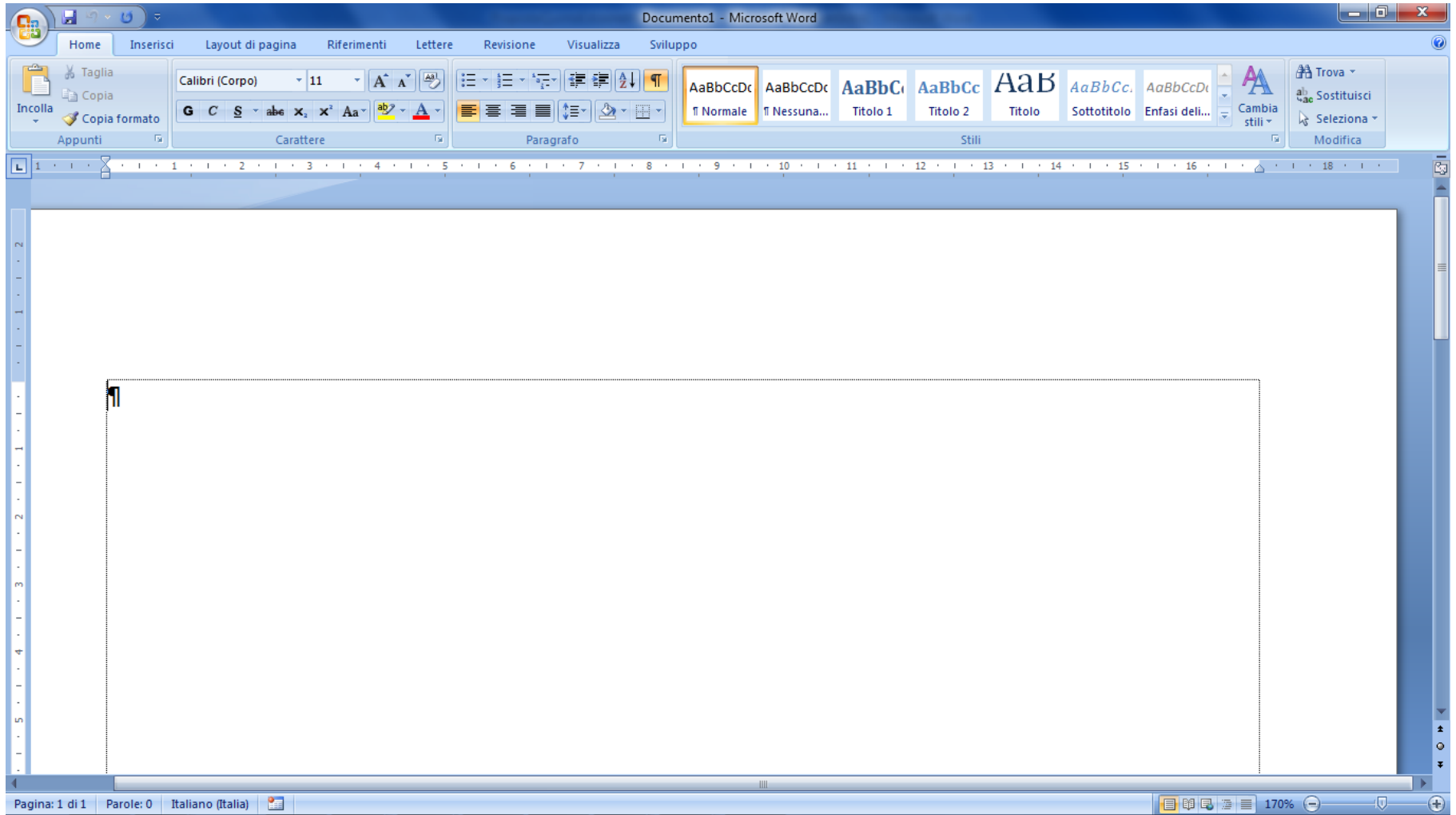
Operating System

Hardware

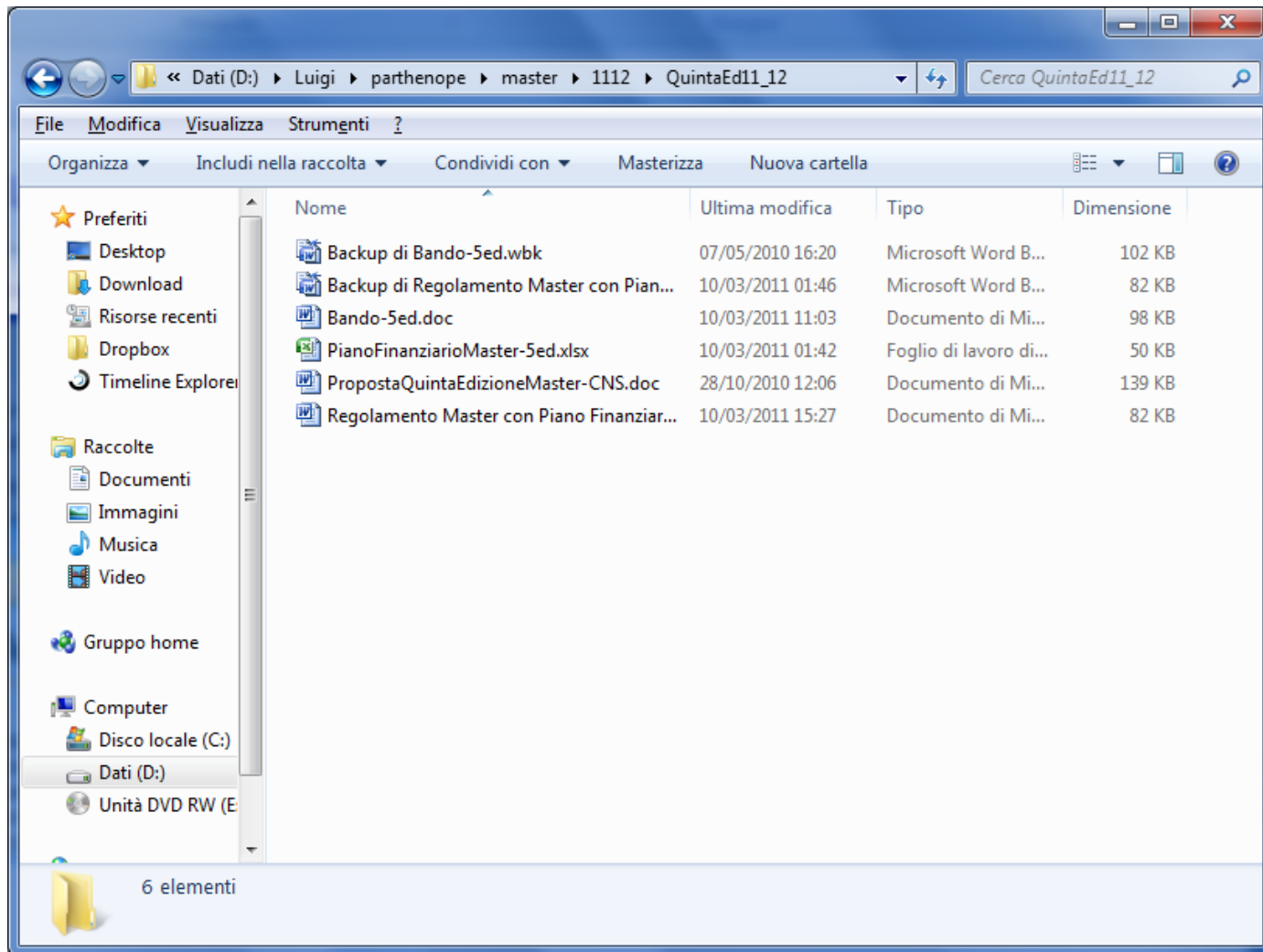
Domande?



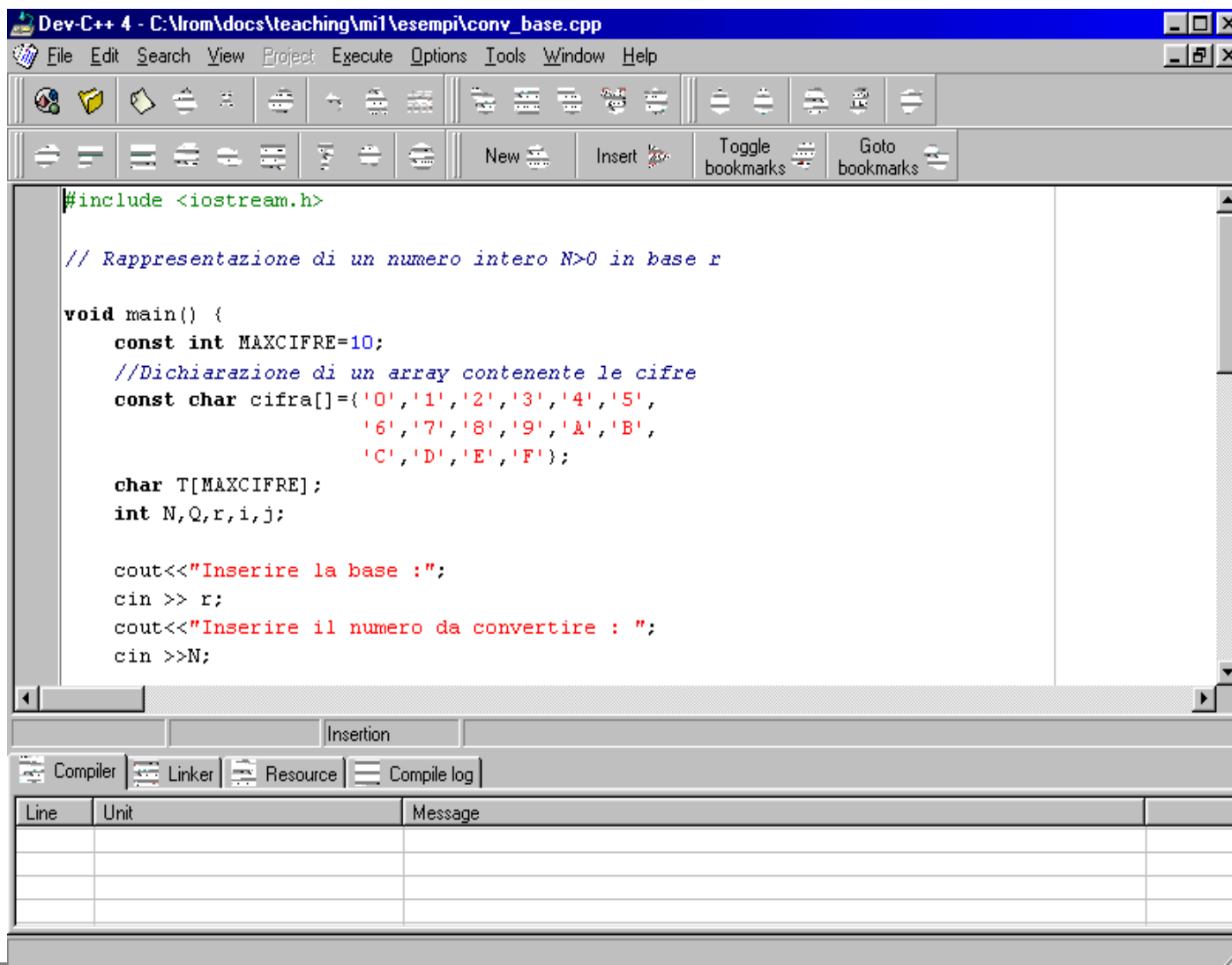
Livello Programmi Applicativi - Interfacce



Livello Programmi Applicativi - Oggetti



Livello Linguaggi di Alto Livello - Interfacce



The screenshot shows the Dev-C++ 4 IDE with a C++ program open. The program is titled "conv_base.cpp" and is located at "C:\rom\docs\teaching\mi1\esempi\conv_base.cpp". The code is as follows:

```
#include <iostream.h>

// Rappresentazione di un numero intero N>0 in base r

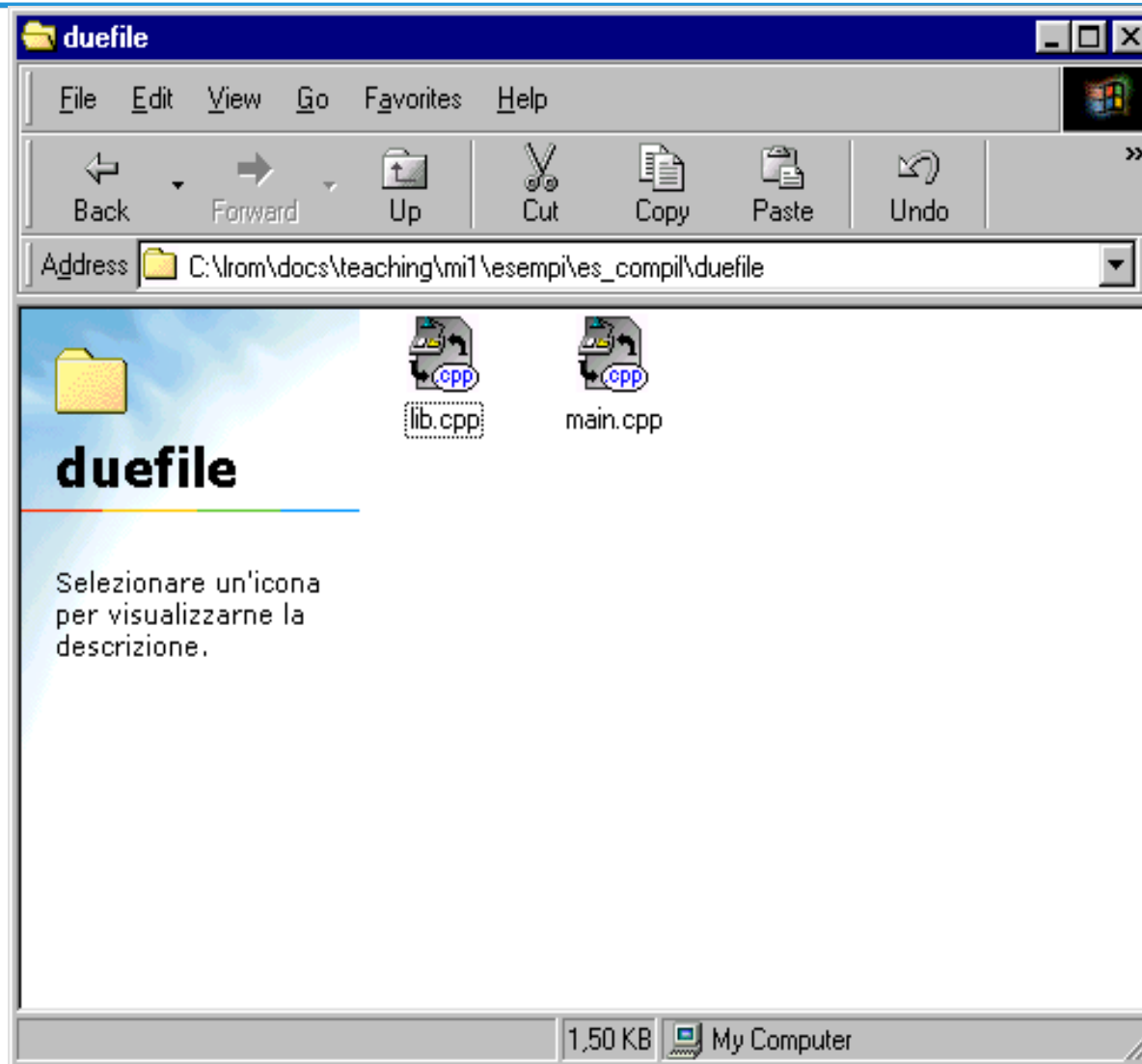
void main() {
    const int MAXCIFRE=10;
    //Dichiarazione di un array contenente le cifre
    const char cifra[10]={'0','1','2','3','4','5',
                        '6','7','8','9','A','B',
                        'C','D','E','F'};

    char T[MAXCIFRE];
    int N,Q,r,i,j;

    cout<<"Inserire la base :";
    cin >> r;
    cout<<"Inserire il numero da convertire : ";
    cin >>N;
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Options, Tools, Window, Help), a toolbar with various icons, and a status bar at the bottom with tabs for Compiler, Linker, Resource, and Compile log. A table with columns for Line, Unit, and Message is visible at the bottom of the IDE window.

Livello Linguaggi di Alto Livello - Oggetti



Livello Assembler - Interfacce

```
alarm.asm - WordPad
File Modifica Visualizza Inserisci Formato ?

;*****
; ALARM.COM
;*****

include mylib.equ

ScEnter equ 1Ch ; scan code del tasto ENTER
BELL equ 07h
TRUE equ 1
FALSE equ 0
blink equ 80h

;-----

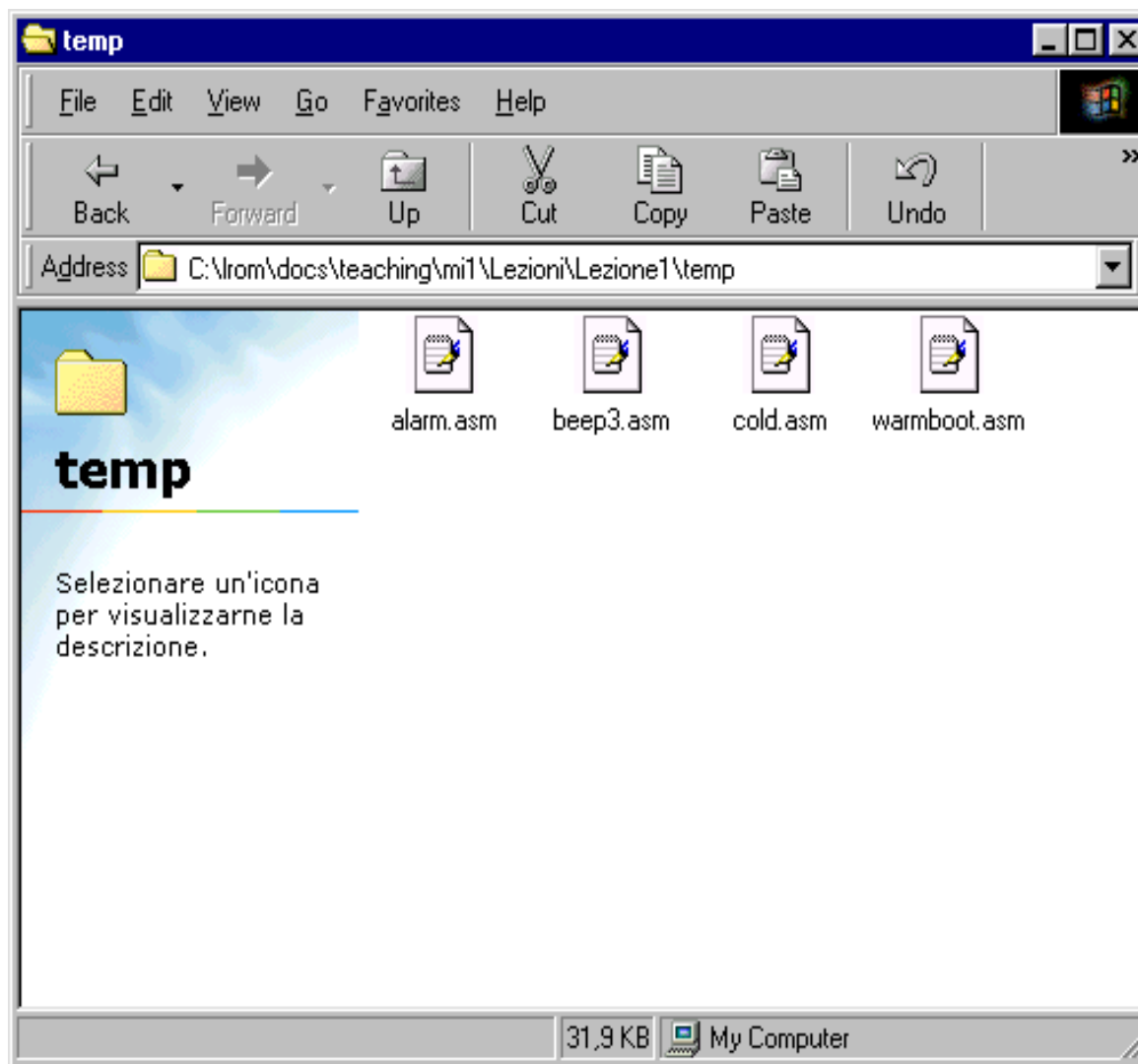
code segment
assume cs:code, ds:code

org 100h

start: jmp main

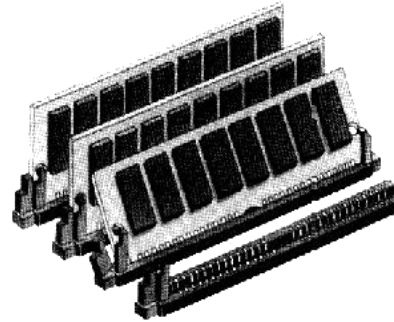
Per aprire la Guida, premere F1.
```

Livello Assembler - Oggetti



Livello Hardware

- Interfacce:
 - Segnali di sincronizzazione dei circuiti
- Oggetti:
 - Componenti Hardware



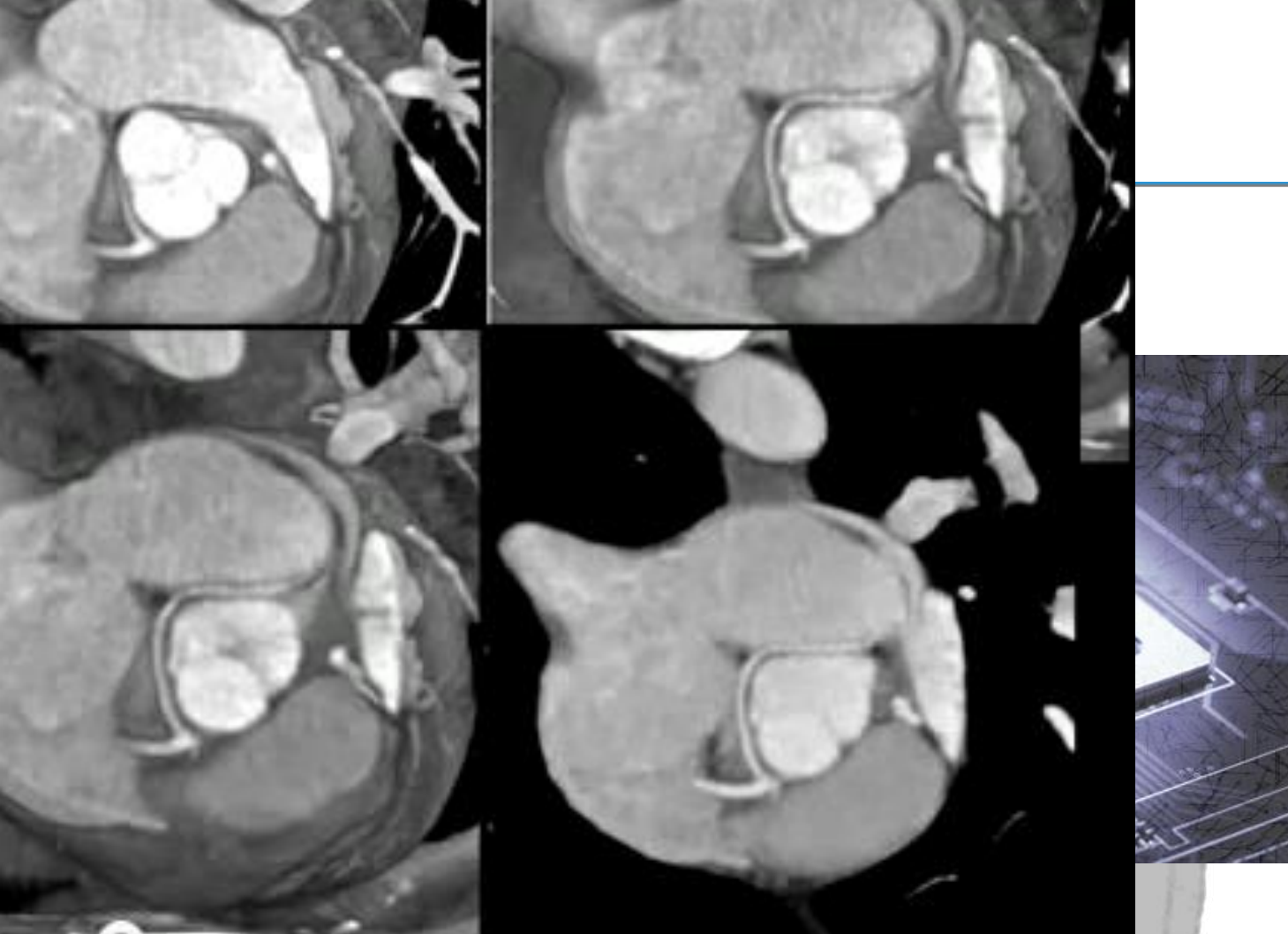
🔔 Nelle Macchine a Controllo Microprogrammato i segnali di sincronizzazione (internamente al processore) sono prelevati da una memoria ROM, mentre nelle macchine a Controllo Cablato, essi sono generati da una rete di controllo

Contesto di Riferimento

- Macchina Assembly
 - File in formato leggibile
 - Formalismo dipendente dall'architettura









The Fault and Intrusion Tolerant NEtworked SystemS (FITNESS) Research Group
<http://www.fitnesslab.eu/>





Contact Info

Contacts



The Fault and Intrusion Tolerant NEtworked SystemS (FITNESS) Research Group
<http://www.fitnesslab.eu/>

