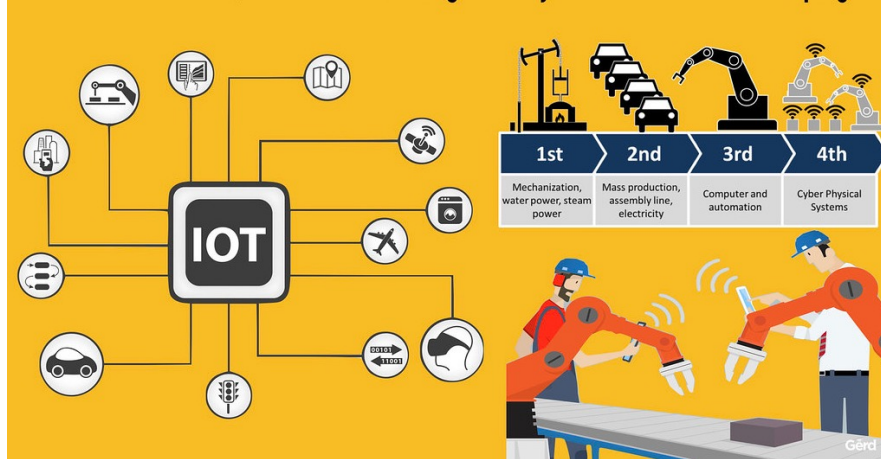


Sistemi Ciberfisici e IoT

martedì 30 marzo 2021



UNIVERSITÀ di VERONA
 Dipartimento di **INFORMATICA**

Agenda

- 14:30 Sistemi Ciberfisici e IoT (Villa/Quaglia)
- 14:50 Argomenti di ricerca del gruppo ESD (Villa/Quaglia)
 - Video 1 Giordano Controls
- 15:00 Presentazione minicorsi (Lora/Fraccaroli/Spellini)
 - Video 2 WAGOO
- 15:20 Opportunità di stage (Bombieri)
 - Video 3 IOTINGA
- 15:30 Dimostrazione live laboratorio ICE (Panato/Tadiello/Bozzini/Chirico)
 - Video 4 EDALAB
- 15:50 Presentazione indirizzo LM Robotics & Smart Industry (Fummi)
 - Video 5 ASEM
- 16:00 Fine

Sistemi Ciberfisici e IoT

Davide Quaglia, Luca Geretti
Tiziano Villa



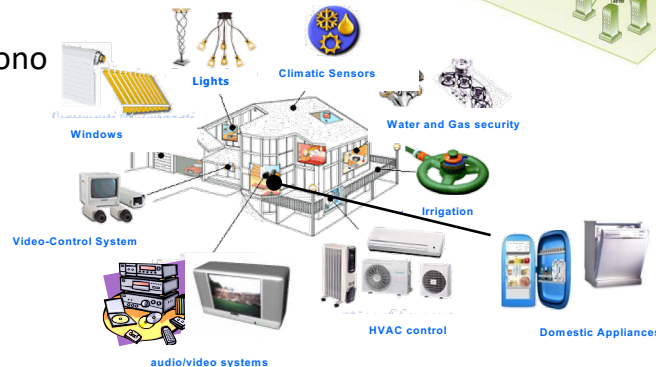
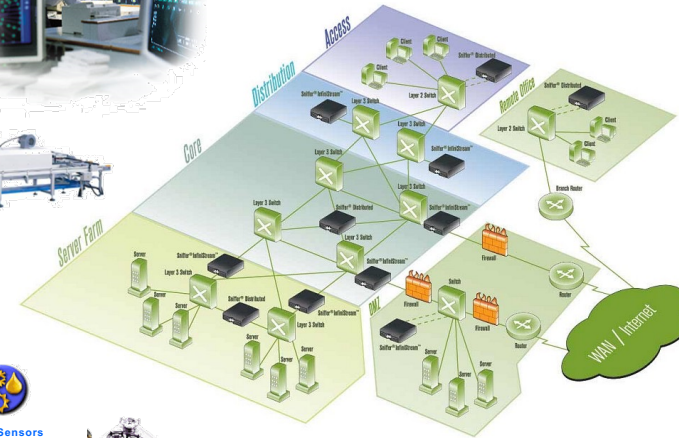
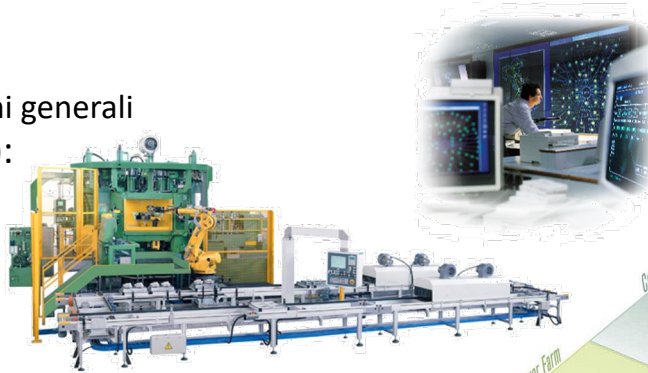
UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**



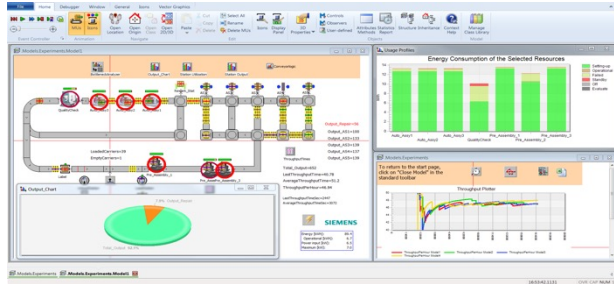
Storia del ciber-universo

- Computer ('60-'80):
 - Sistemi per la soluzione di problemi generali
- Sistemi di controllo digitali ('80-'90):
 - Automazione
- Sistemi distribuiti ('90-'00):
 - Computer connessi in rete
 - Esplosione del fenomeno Internet
- Sistemi embedded ('00-):
 - Sistemi immersi nell'ambiente
- **Sistemi ciberfisici ('10-):**
 - Sistemi embedded distribuiti che interagiscono col mondo fisico
- **Internet of Things (IoT) ('20-):**
 - Oggetti fisici digitalizzati e interconnessi

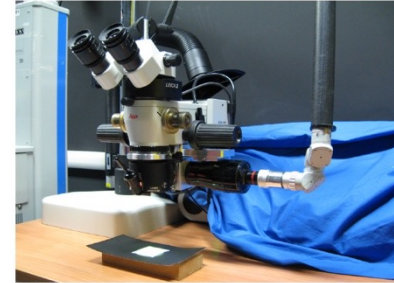


Sistemi ciberfisici

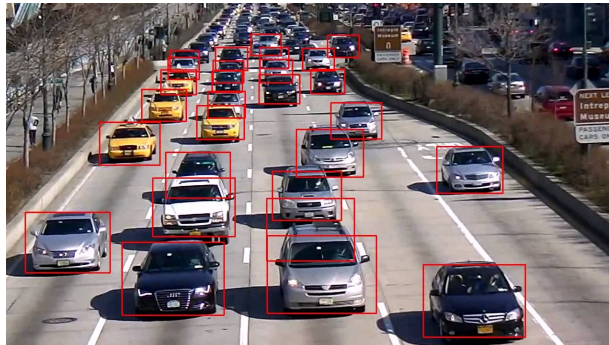
Industry 4.0



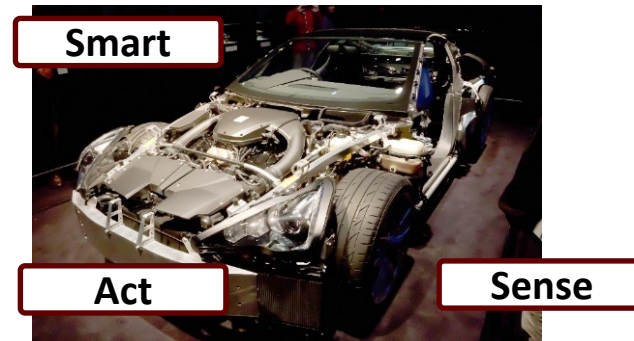
Robotic surgery



Embedded vision

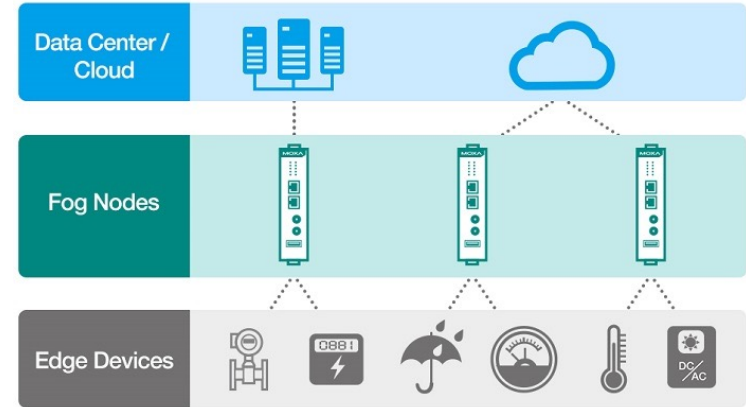


Automotive

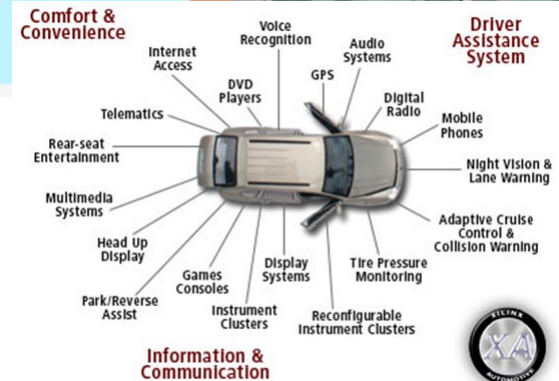
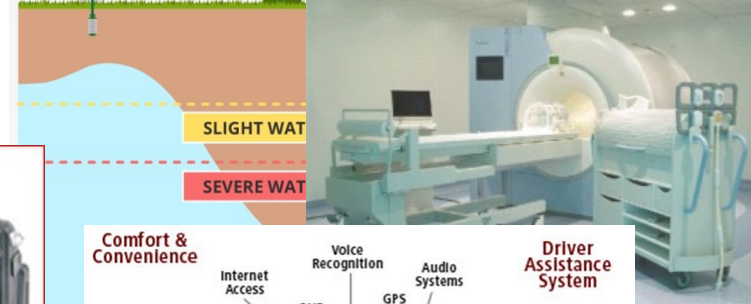
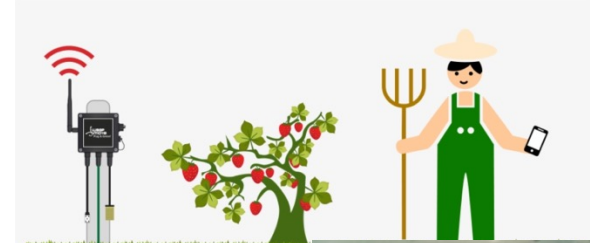
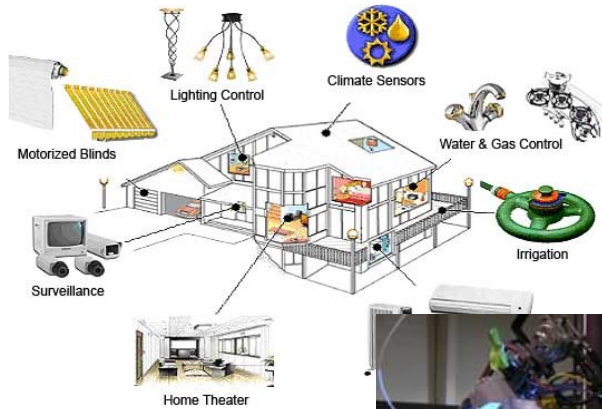


Dal sensore al cloud attraverso l'edge/fog computing

- Il dispositivo sensore/attuatore è solo un pezzo del panorama IoT
- Il dato può essere elaborato
 - vicino al dispositivo -> **Edge** computing
 - al centro della rete -> **Cloud** computing
 - in mezzo -> **Fog** computing
- Nuove architetture di rete come 5G e LoRaWAN sono pesantemente impostate in questa ottica
 - Come lo smartphone ha creato il mercato delle APP, queste nuove architetture creeranno nuove opportunità per programmare servizi a questi tre livelli
 - Sviluppo aperto e non monopolizzato dai venditori di apparati e dai gestori della rete



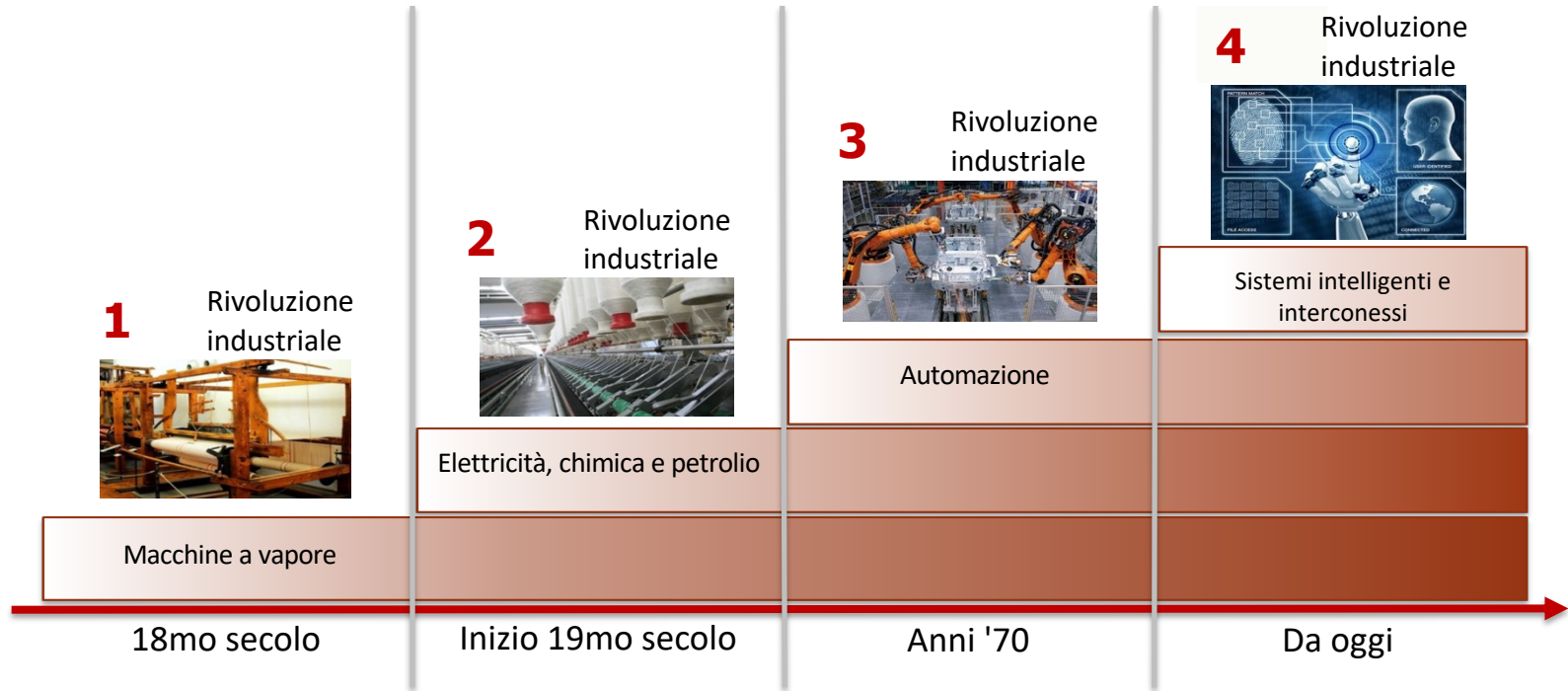
Sistemi ciberfisici e IoT: dove?



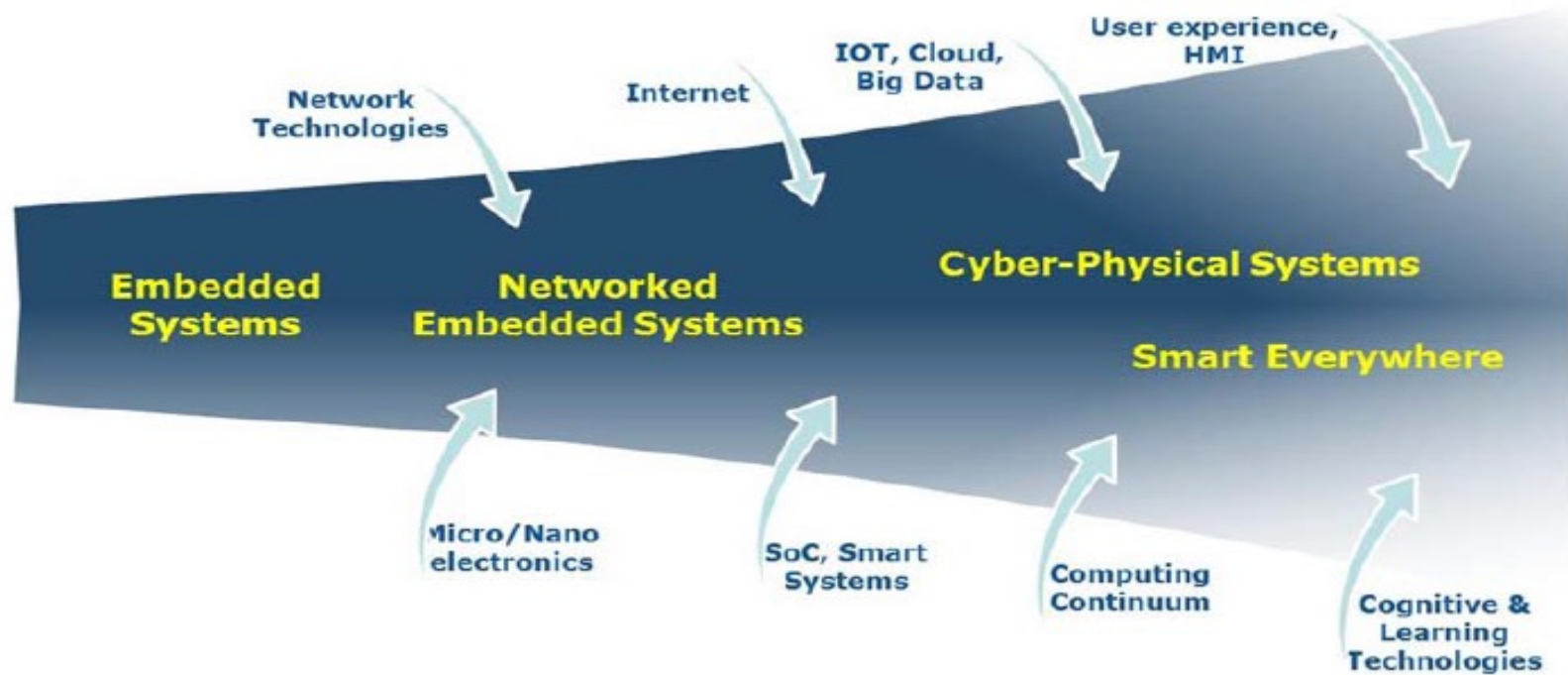
Importanza economica



Importanza economica: Industria 4.0



Discipline coinvolte



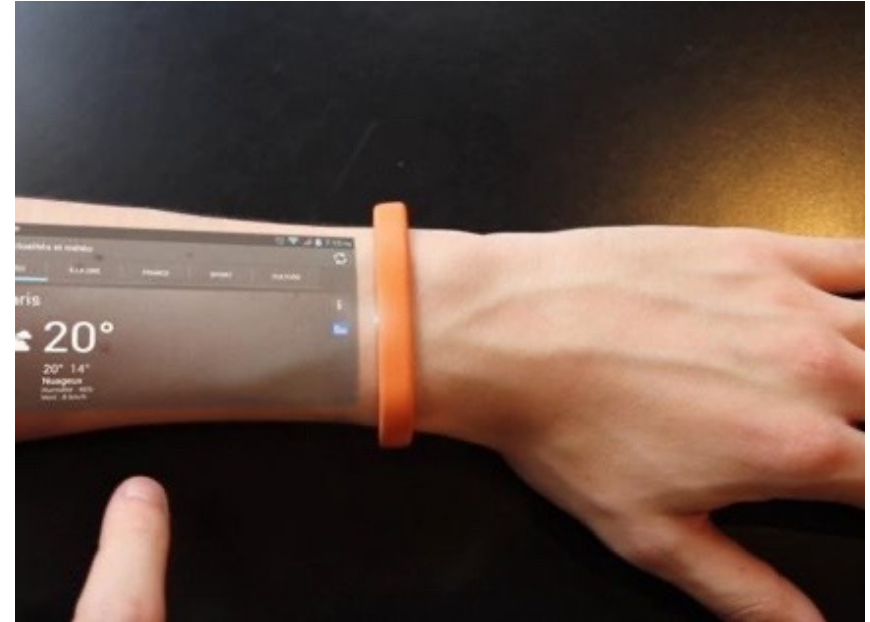
IoT: come progettarlo?

- Non possiamo progettare un sistema IoT come si progettava un sistema di calcolo generale
 - Vincoli diversi, obiettivi diversi
 - Visione unificata di hardware, software e rete
 - Anche l'ambiente circostante deve essere tenuto in considerazione



Vincoli di progetto

- Dimensioni e peso
- Energia
- Robustezza all'ambiente
 - Calore, umidità, interferenze, vibrazioni, ...
- Azione in tempo reale
- Affidabilità
- Bassi costi



Competenze richieste

- Metodi formali: progettazione basata su modelli matematici
- Architetture hardware
 - bilanciamento corretto tra HW e SW
- Programmazione
 - nuovi linguaggi
- Rete
 - le comunicazioni sono fondamentali
 - moltissime architetture di rete tra cui scegliere
 - nuove reti --> nuove opportunità
- Confluenza delle 3C: **C**omputazione, **C**ontrollo e **C**omunicazione

Argomenti di ricerca del gruppo ESD

Davide Quaglia



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**



Mapping

- Design
 - Synthesis of **network architectures** (Quaglia-Fraccaroli)
 - Embedded **vision** applications (Bombieri-Aldegheri)
 - **Edge**-computing (Bombieri-Aldegheri)
- Modeling & Verification
 - Analysis of industrial **plants & processes** (Fummi-Centomo-Spellini)
 - Joint system-network **simulation** (Quaglia)
 - Catching sources of **vulnerabilities** (Pravadelli-Germiniani)
 - Automatic **analog** abstraction (Fummi-Fraccaroli)
 - **ROS**-compliant containerized verification monitors (Pravadelli-Bombieri-Germiniani)
 - Run-time verification of **parametric cyber-physical systems** (Villa-Geretti)

Attività di tipo D

mini-corsi

Enrico Fraccaroli
Michele Lora
Stefano Spellini
Nicola Dall'Ora



UNIVERSITÀ
di **VERONA**

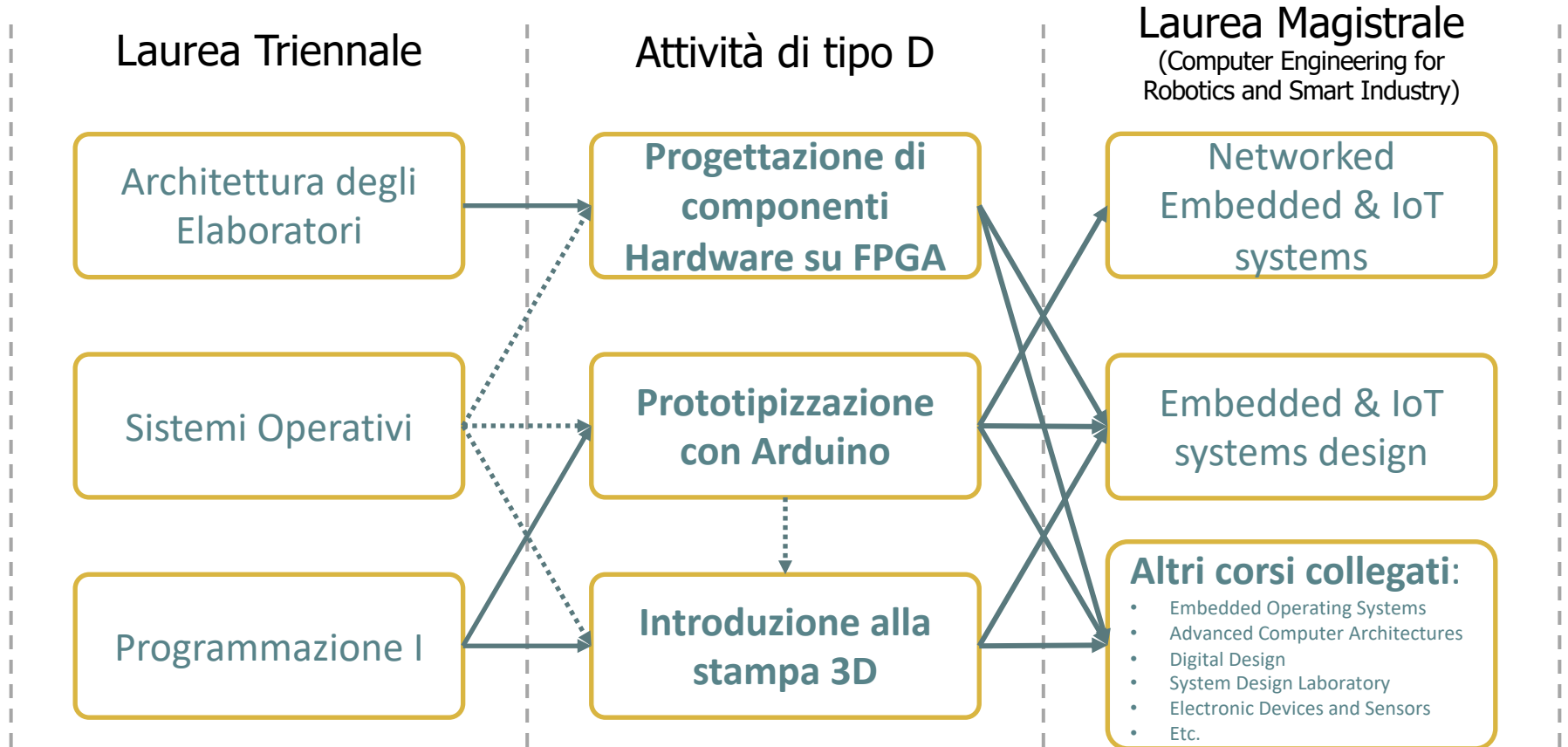
Dipartimento
di **INFORMATICA**



Overview

- Tre corsi per **toccare con mano** i sistemi embedded:
 - Progettazione di componenti **Hardware su FPGA**
 - Prototipizzazione con **Arduino**
 - Introduzione alla **stampa 3D**
- **Obiettivi:**
 - Introdurre all'utilizzo di strumenti per l'implementazione di sistemi concreti
 - Fornire un «piccolo vantaggio» per la Laurea Magistrale in Computer Engineering for Robotics and Smart Industry

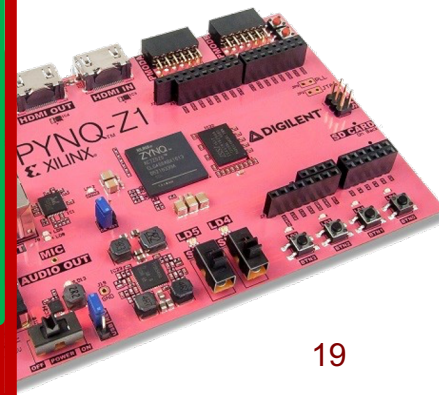
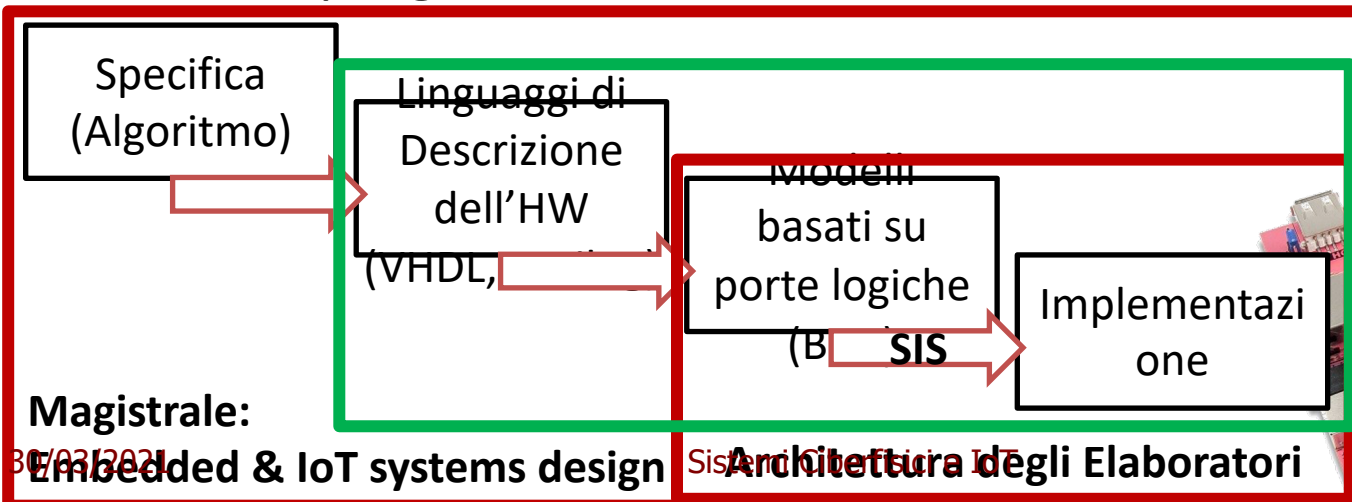
Contesto dei corsi



Progettazione Hardware su FPGA

Introduzione Generale

- FPGA: Field-Programmable Gate Array
 - Circuito integrato progettato per esser riconfigurato e customizzato
 - Prototipizzazione di **circuiti integrati**
 - Piattaforma per implementare sistemi embedded ed IoT ad **alte prestazioni**
 - Amazon AWS Elastic Cloud: computazione accelerata su FPGA
- Flusso di progettazione di Hardware



Progettazione Hardware su FPGA

Dettagli, Prerequisiti ed Obiettivi

- **Dettagli**
 - Studenti: 15
 - Durata: 8 ore
 - Periodo: fine aprile – inizio maggio
 - FPGA disponibili: Xilinx Pynq-Z1 con ARM Cortex-A9 MPCore SoC, 1.3M gates riconfigurabili
- **Prerequisiti**
 - Architettura degli Elaboratori
 - Programmazione ed elementi di Sistemi Operativi
- **Obiettivi del corso: introduzione alla progettazione di dispositivi embedded**
 - Concetti di base di Verilog
 - Principi di progettazione hardware in Verilog
 - Sintesi automatica di progetti hardware
 - Principi di co-progettazione hardware/software

- Progetto a scelta dello studente:
 - Tra quelli proposti dal docente
 - Proposto dallo studente e concordato con il docente
- Esempio di progetto proposto:
Progettazione ed implementazione di un componente hardware per velocizzare il **calcolo della moltiplicazione di matrici**
 - Identificazione dell'algoritmo da utilizzare
 - Progettazione dell'architettura del componente hardware
 - Implementazione del componente in Verilog
 - Sintesi automatica del componente, deploy su FPGA, test del dispositivo e valutazione delle prestazioni

Prototipazione con Arduino

Introduzione Generale

- Cos'è Arduino?
 - E' una piattaforma di prototipazione che comprende un microcontrollore e una serie di interfacce sia analogiche che digitali
- A cosa serve Arduino?
 - E' una piattaforma per la prototipazione rapida di progetti che richiedono l'uso di componenti: elettroniche, meccaniche, comunicazioni, idraulica
- Perché Arduino?
 - È bello...
 - Costa poco, permette di avvicinarsi all'elettronica, è facile da utilizzare e ha una community molto attiva
 - Viene utilizzato in: hobbistica, insegnamento e ambito professionale



Prototipazione con Arduino

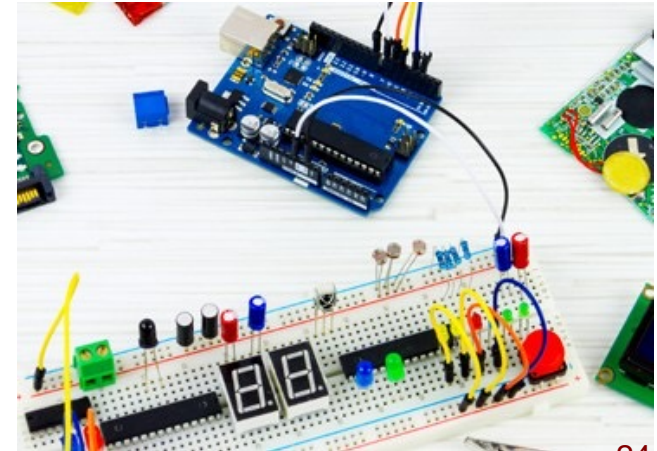
Dettagli, Prerequisiti e Obiettivi

- **Dettagli**
 - Studenti: 15 *max*
 - Durata: 8 ore
 - Periodo: Giugno/Luglio
 - Prima del corso su stampa 3D
 - Arduini e componenti disponibili durante il corso: Arduino UNO, componenti elettriche e sensoristica
- **Prerequisiti**
 - Conoscenza base del linguaggio C (Programmazione I) o C++
 - Conoscenza base della fisica (Fisica I)
- **Obiettivi del corso: prototipazione con Arduino**
 - Introduzione ai concetti di base di elettronica
 - Introduzione alla cross-compilazione
 - Progettazione e sviluppo di semplici circuiti elettrici
 - Introduzione all'edge-computing e all'invio dei dati su cloud

Prototipazione con Arduino

Modalità d'Esame

- L'esame consisterà in un **progetto a scelta**:
 - Tra un insieme di progetti proposti
 - Libero dello studente (da concordare)
- Esempio di progetto proposto:
design e **implementazione** di un sistema intelligente di irrigazione
 - Sviluppo in ambiente Arduino (codice sorgente)
 - Testing del corretto funzionamento (relazione)
 - Discussione delle scelte implementative (esame orale)

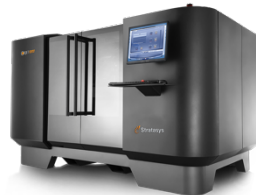


Introduzione Stampa 3D

Introduzione Generale

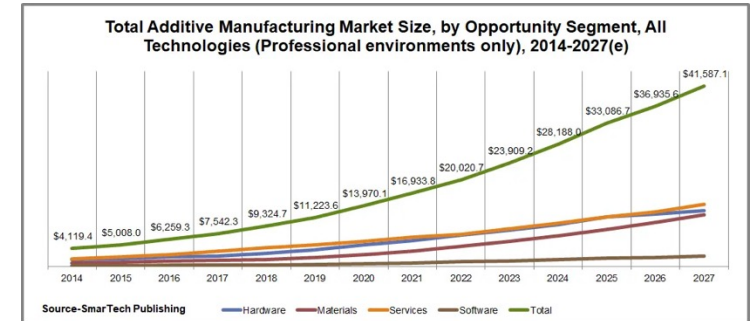
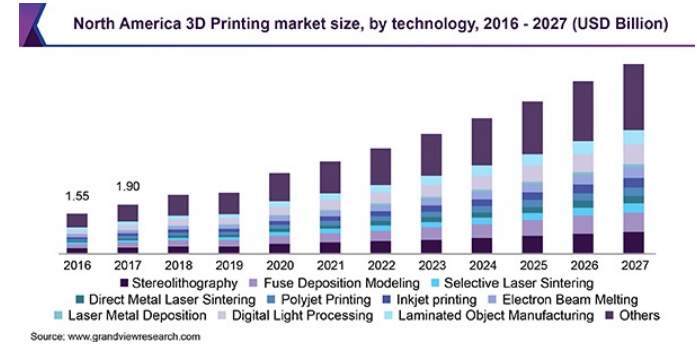
Manifattura additiva e stampa 3D: processo di costruzione automatica di oggetti, strato dopo strato, partendo da dati digitali

- Benefici principali:
 - Flessibilità
 - Sostenibilità
 - Semplificazione processo produttivo
- Pilastro fondamentale dell'Industria 4.0
- Mercato in crescita costante
- Competenze richieste nell'industria moderna



30/03/2021

Sistemi Ciberfisici e IoT



25

Introduzione Stampa 3D

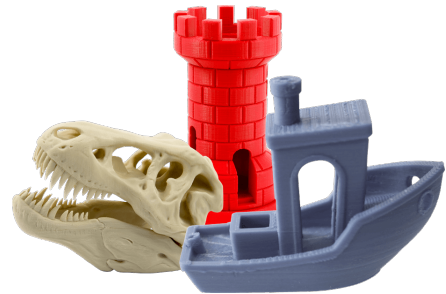
Dettagli, Prerequisiti e Obiettivi

- **Dettagli**
 - Studenti: *10 max*
 - Durata: *8 ore*
 - Periodo: Giugno/Luglio
 - Dopo il corso di prototipizzazione su Arduino
 - Stampanti disponibili durante il corso agli studenti: *Prusa3D MK3S+*
- **Prerequisiti**
 - Conoscenza base del linguaggio C (Programmazione I)
 - Conoscenza base della geometria Euclidea
 - Conoscenza base dell'ambiente di sviluppo Arduino (opzionale)
- **Obiettivi del corso: introduzione alla stampa 3D *Fused Deposition Modeling* (FDM) e al firmware open-source Marlin**
 - Introduzione ai concetti di base (sistema riferimento cartesiano, GCODE, slicing, etc.)
 - Configurazione processo di stampa
 - Configurazione firmware
 - Design e integrazione nuove funzionalità

Introduzione Stampa 3D

Modalità d'Esame

- L'esame consisterà in un **progetto a scelta**:
 - Tra un insieme di progetti proposti
 - Libera dello studente (da concordare)
- Esempio di progetto proposto:
design e implementazione di una **nuova funzionalità** a livello firmware sulla stampante Prusa3D MK3S+
 - Sviluppo in ambiente Arduino (codice sorgente)
 - Testing del corretto funzionamento (relazione)
 - Discussione delle scelte implementative (esame orale)



Modalità di iscrizione ai corsi

- Modulo (motivazionale) da inviare al docente
- Finestra iscrizioni:
 - Progettazione hardware su FPGA: prima metà di Aprile
 - Prototipizzazione con Arduino: Maggio/Giugno
 - Introduzione alla stampa 3D: Giugno

Opportunità di stage

Nicola Bombieri



UNIVERSITÀ
di **VERONA**

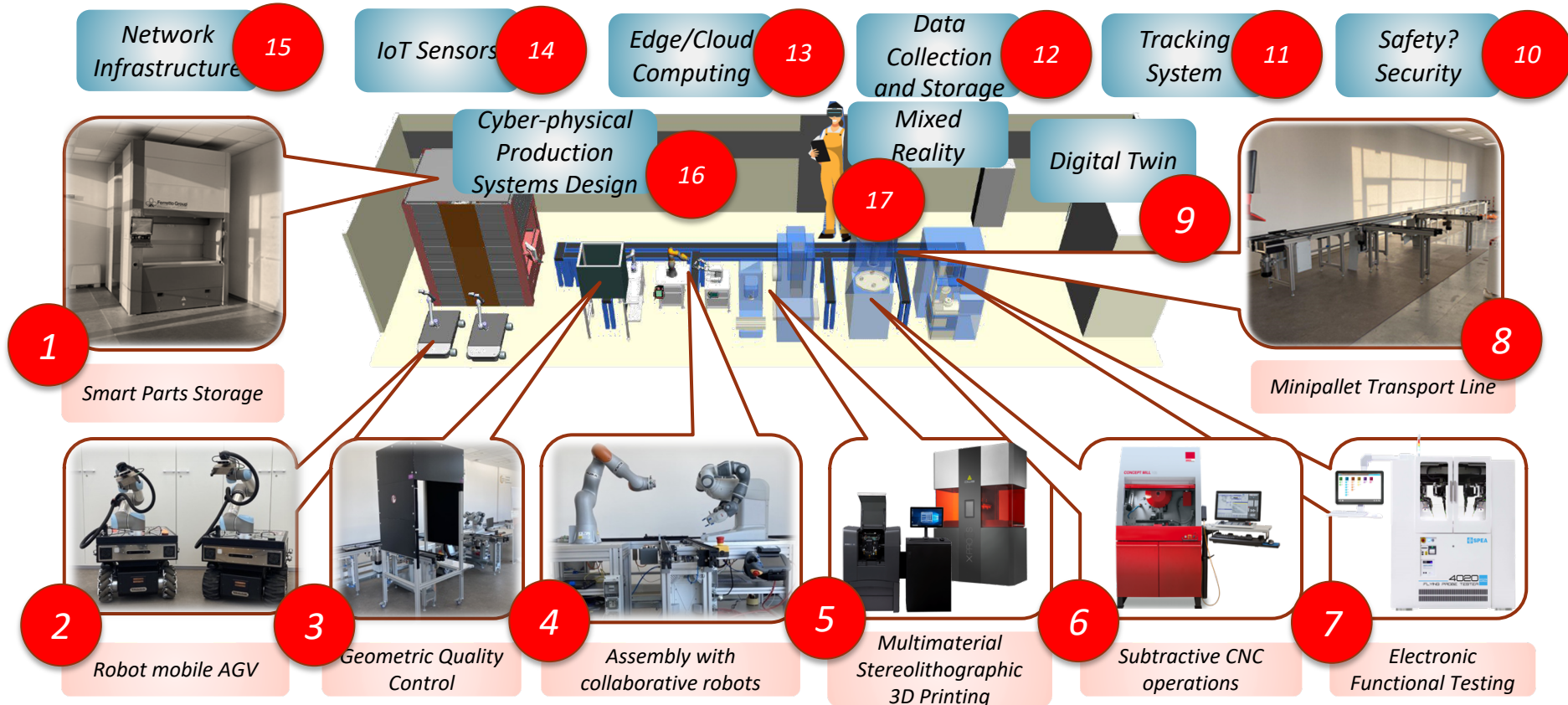
Dipartimento
di **INFORMATICA**



Quando, dove, perchè

- Sia per LT che LM
- ≥ 6 CFU (25ore/CFU)
- Spesso prima di inizio tesi
 - Possibilità di estensione in tesi
- In:
 - ICE lab
 - PARCO lab
 - ESD lab
 - Aziende associate (rif. video)
- Perchè:
 - Esperienza in CV extra curriculum didattico
 - Introduzione ad ambiente di progettazione avanzato

Industrial Computer Engineering (ICE) Lab.



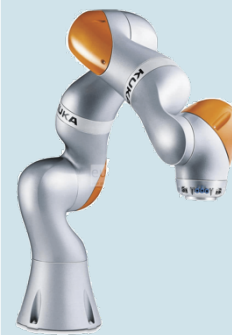
Logistic 4.0

Machine	TODO	Skills	Degree		Remote work			Effort
			Bachelor	Master	Yes	Partial	No	
	PLC software expansion: Add new features, fix bugs	PLC Siemens		X			X	12
	HMI expansion: Add new command and improve graphics	Graphics	X			X		3
	Move arm with joystick	Programming Robotics ROS		X			X	12
	Extend ICE demo program with new tasks	Programming Robotics ROS		X			X	12
	Logger to data collection via Rest-API or MQTT	Programming	X	X		X		6

Visual control

Machine	TODO	Skills	Degree		Remote work			Effort
			Bachelor	Master	Yes	Partial	No	
Quality controls 	Program expansion: piece recognition	Programming Visual	X			X		6
Robot for quality controls 	Pick and move arm: pick part and move over camera		X	X		X		6

Assembling and disassembling

Machine	TODO	Skills	Degree		Remote work			Effort
			Bachelor	Master	Yes	Partial	No	
	Assembly task	Programming Robotics		X			X	12
	Logger to data collection via Rest-API or MQTT (performance test)	Programming	X	X		X		6
	Create OPC-UA server	Programming	X	X		X		6
	Assembly task	Programming Robotics		X			X	12
	Install OPC-UA server on Kuka	Programming Robotics	X	X			X	6
	Define unique interface for the 3/4 ros KUKA stacks	Programming Robotics ROS		X		X		12
	Rewrite drivers realtime stack ros KUKA	Programming Robotics		X		X		12
	Logger to data collection via Rest-API or MQTT (performance test)	Programming	X	X		X		6
	Learn a recipe from Kuka's movements	Programming Robotics		X			X	12

30/03/2021

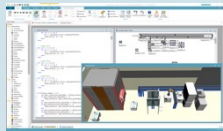
Sistemi Cibernetici e IoT

34

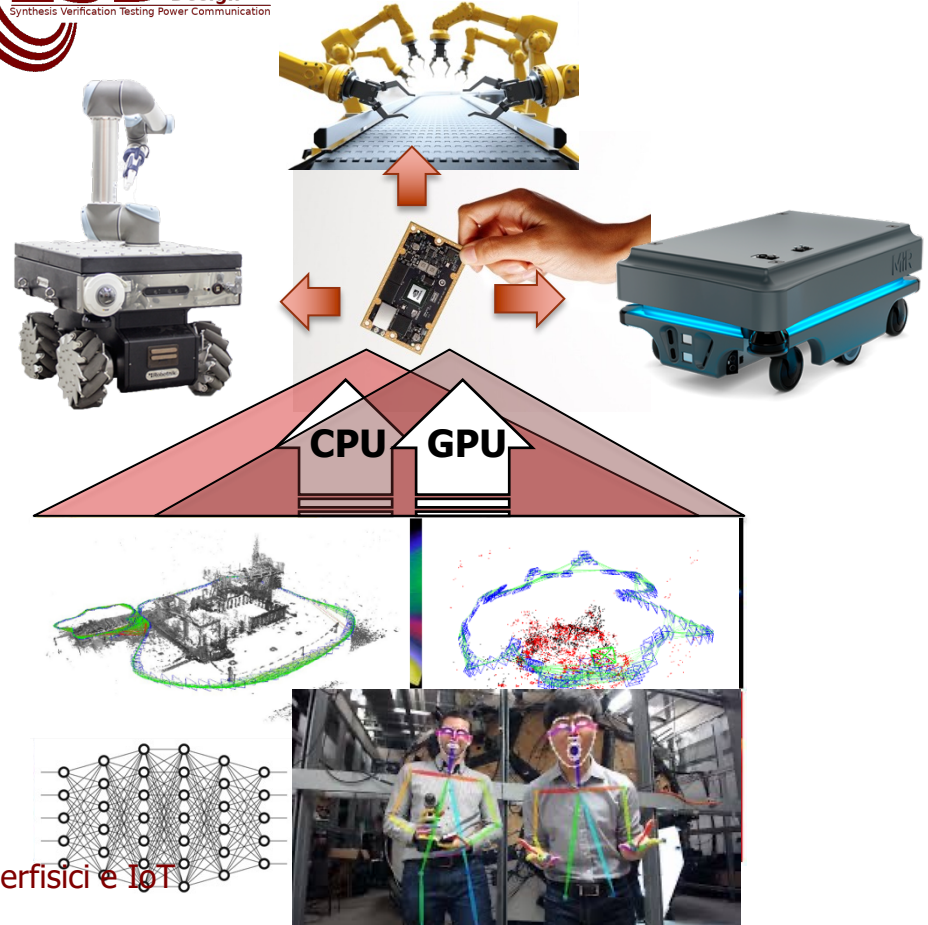
Additive synthesis

Machine	TODO	Skills	Degree		Remote work			Effort
			Bachelor	Master	Yes	Partial	No	
DWS 3D Printers 	Create opcua server, logger to data collection via Rest-API or MQTT	Programming	X		X			6
Stratasys 3D Printers 	Create opcua server, logger to data collection via Rest-API or MQTT	Programming	X		X			6
Stratasys and DWS 3D Printers	Insert in the demo the video of the print. Assembly and editing the video.		X				X	2

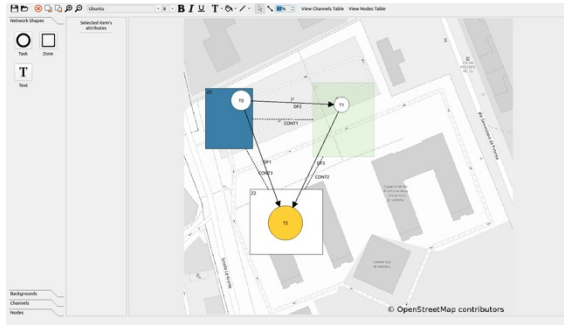
IoT and IIoT data gathering

Machine	TODO	Skills	Degree		Remote work			Effort
			Bachelor	Master	Yes	Partial	No	
IoT/IIoT 	Dashboard expansion (UNIQUO)	Programming Graphics	X		X			4
	Dashboard implementation (Grafana)	Programming Graphics	X		X			4
	Duemmegi(light control unit) interface expansion: control light module expansion	Programming	X			X		6
Digital twin 	Improve the automatic generation flow	Theoretical Programming		X	X			6

- **Programming** of heterogeneous embedded devices for **edge computing**
 - Advanced SW programming targeting *performance* and *energy efficiency*
 - Multi-core CPU/Many-core (GPU) parallel programming
 - AI computing at the edge
 - Intelligent data processing at the edge
 - Wearable (IoT) devices programming
 - Network design and verification

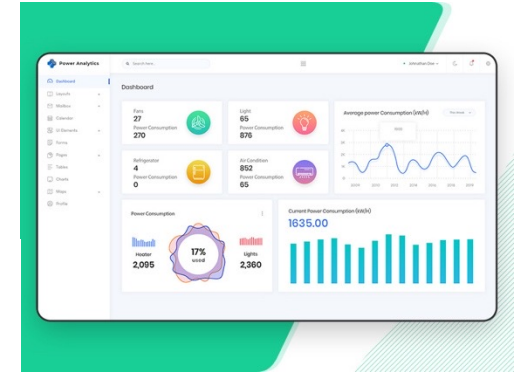
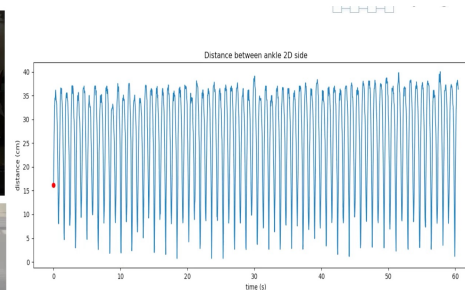
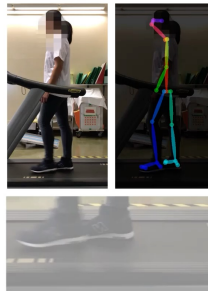


• Esempi:



Estensione di una GUI in Python per la descrizione di scenari di rete IoT

Ottimizzazione codice C/C++/CUDA e sviluppo interfaccia pose estimation



Sviluppo di applicazioni web-based per la raccolta, visualizzazione e analisi di dati da sorgenti IoT

Industrial Computer Engineering Lab.

a smart factory for didactics – the ESD view

Franco Fummi



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**



Computer Engineering for Robotics and Smart Industry

Objective of the course

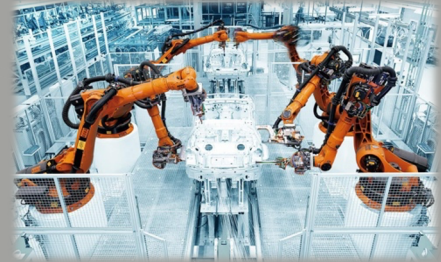
Dominate ICT technology for the *design, integration and management* of industrial production plants

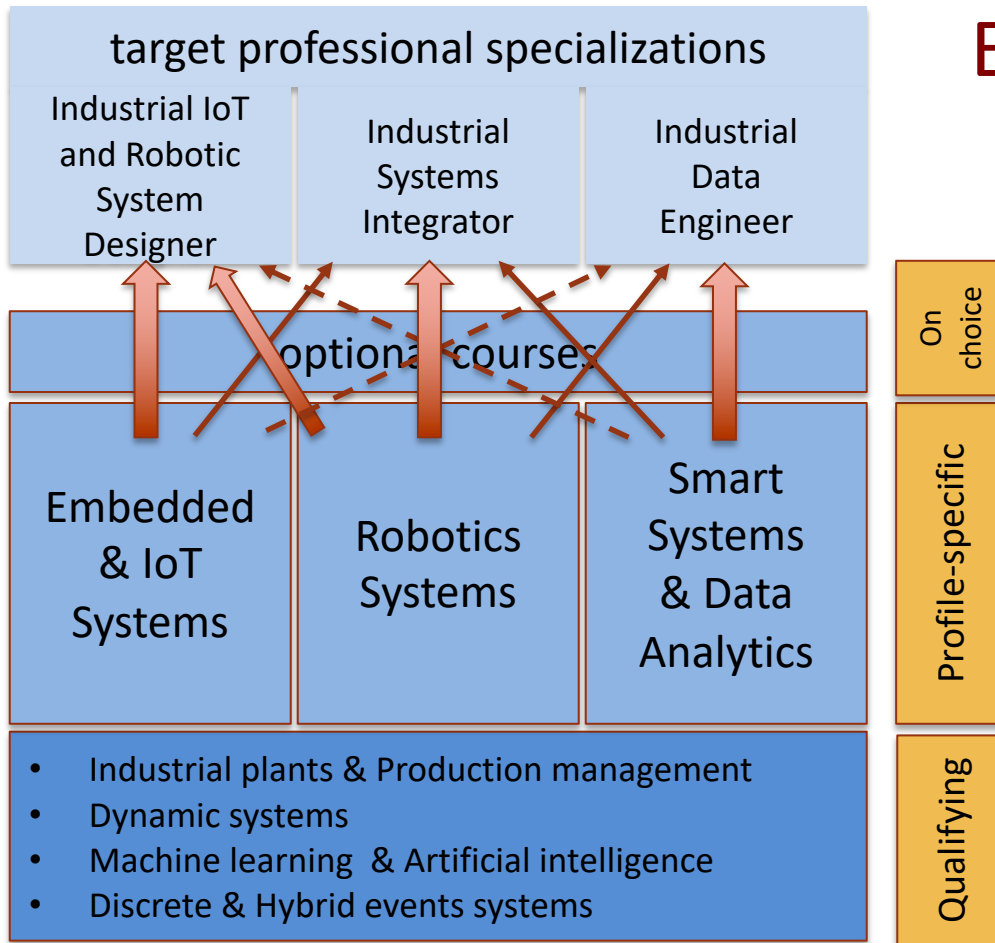
Target professional figures

Industrial IoT and Robotic Systems Designer
Industrial Systems Integrator
Production Programmer and Analyst

Strategic skills

Industrial robotics
Cyber-physical systems
Data analytics
Digital manufacturing

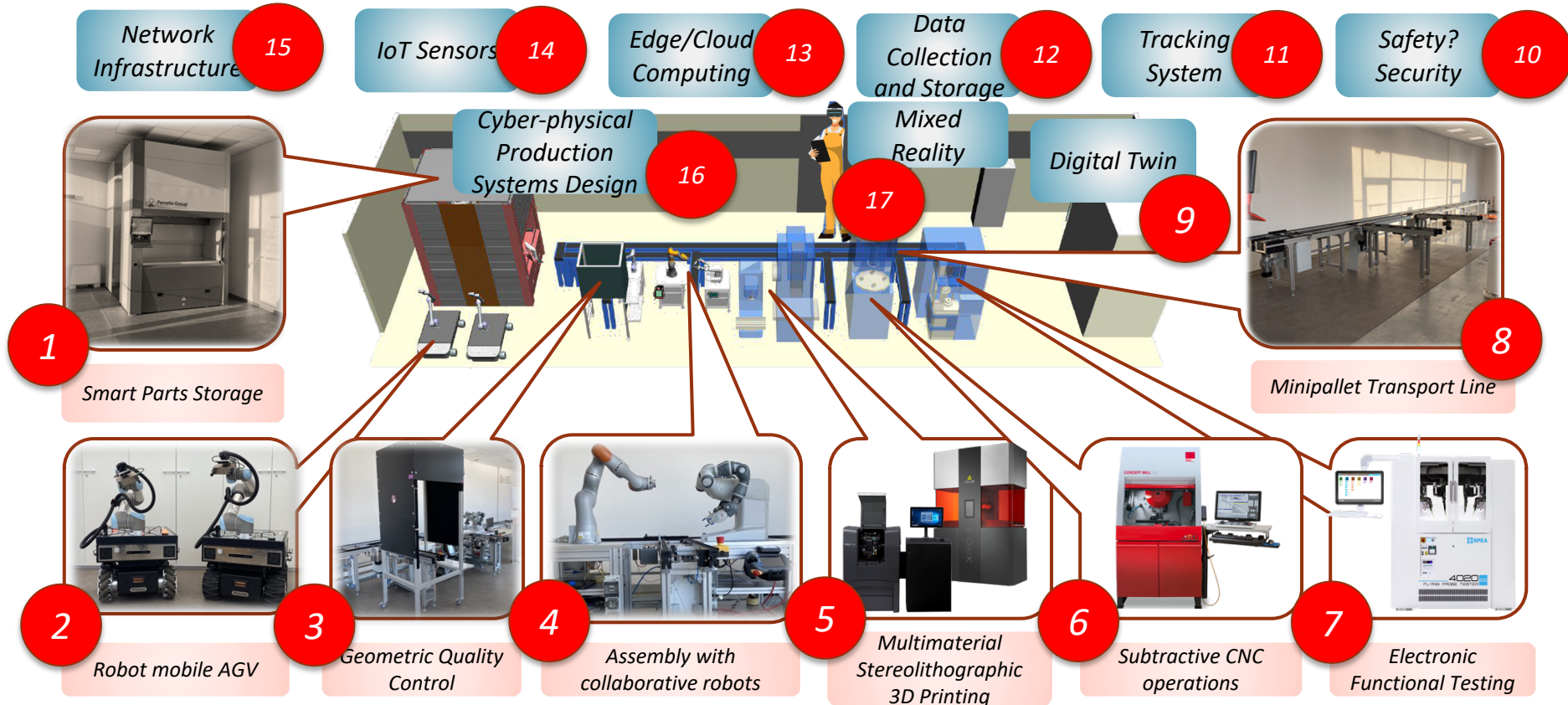




Exams Organization

- Qualifying exams are mandatory
- One profile must be completed
- A target professional specialization must be reached by following:
 - other profile-specific exams
 - On choice exams
 - coherent exams from other master degrees

A completely reconf. production line



- Discrete event and hybrid systems – Villa

QUALIFYING

Discrete-Event and Hybrid Systems - Theory

- Tiziano Villa
- Formal methods for the specification, analysis and synthesis of discrete event and hybrid systems :
 - Languages, automata and state machines
 - Petri nets
 - Supervisory control for discrete event systems
 - Scheduling and planning
 - Hybrid systems
 - Assume-guarantee contracts

Discrete and Hybrid Systems - Contributions

This class introduces the theoretical foundations to be applied in the successive Systems Design Laboratory, so that together they describe the ideas and tools that enable model-based design of cyber-physical production systems **(16)**

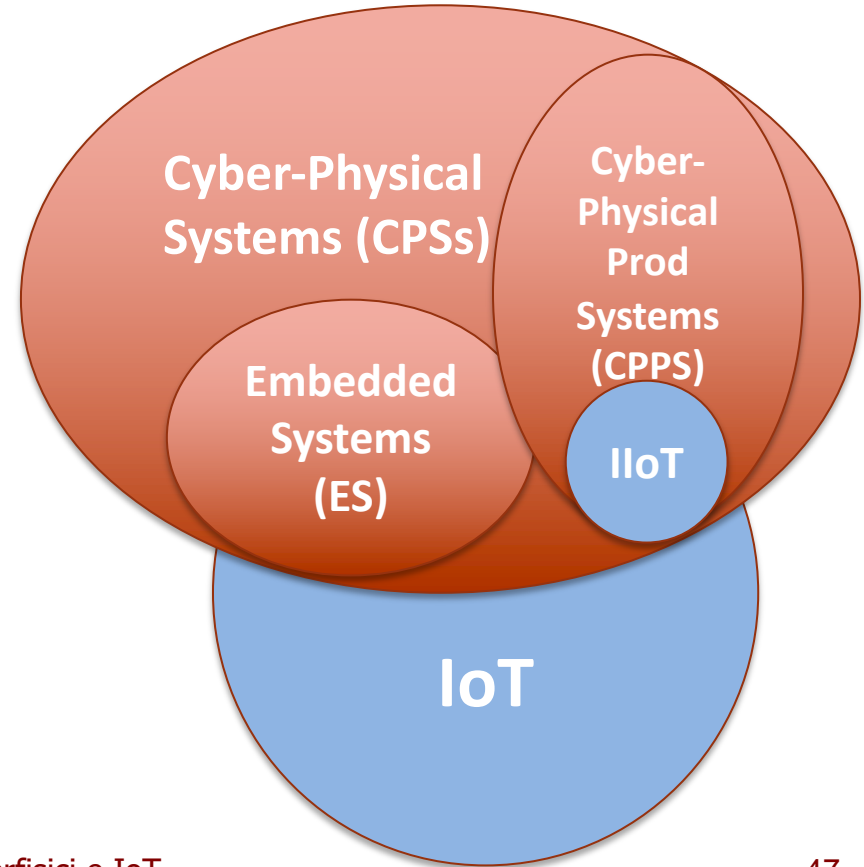
- At the end, the student will understand the theoretical foundations of formalisms and algorithms to specify, analyze and synthesize discrete-event and hybrid systems:
 - represent discrete-event and hybrid systems by means of languages, finite-state automata and machines, Petri nets, hybrid automata
 - analyze their behaviour by formal methods (structural and behavioural, exact and approximate)
 - synthesize supervisory controllers of plants described by finite automata with uncontrollable and unobservable events
 - analyze the behaviour of simple hybrid systems with continuous and discrete dynamics
 - organize the design process as a flow of correct-by-construction transformations

- Embedded and IoT Systems Design - Fummi
- Advanced Computer Architectures - Bombieri
- Networked Embedded & IoT Systems - Quaglia
- Embedded Operating Systems – Pravadelli

PROFILE: EMBEDDED AND IOT SYSTEMS DESIGN

Embedded and IoT Systems Design - view

- Franco Fummi
- Methodologies and tools for modelling and design of CPPSs and IIoT
 - hw/sw modelling
 - hw/sw design
 - IoT sensor-edge-cloud infrastructure
 - CPPs modeling, simulation, integration

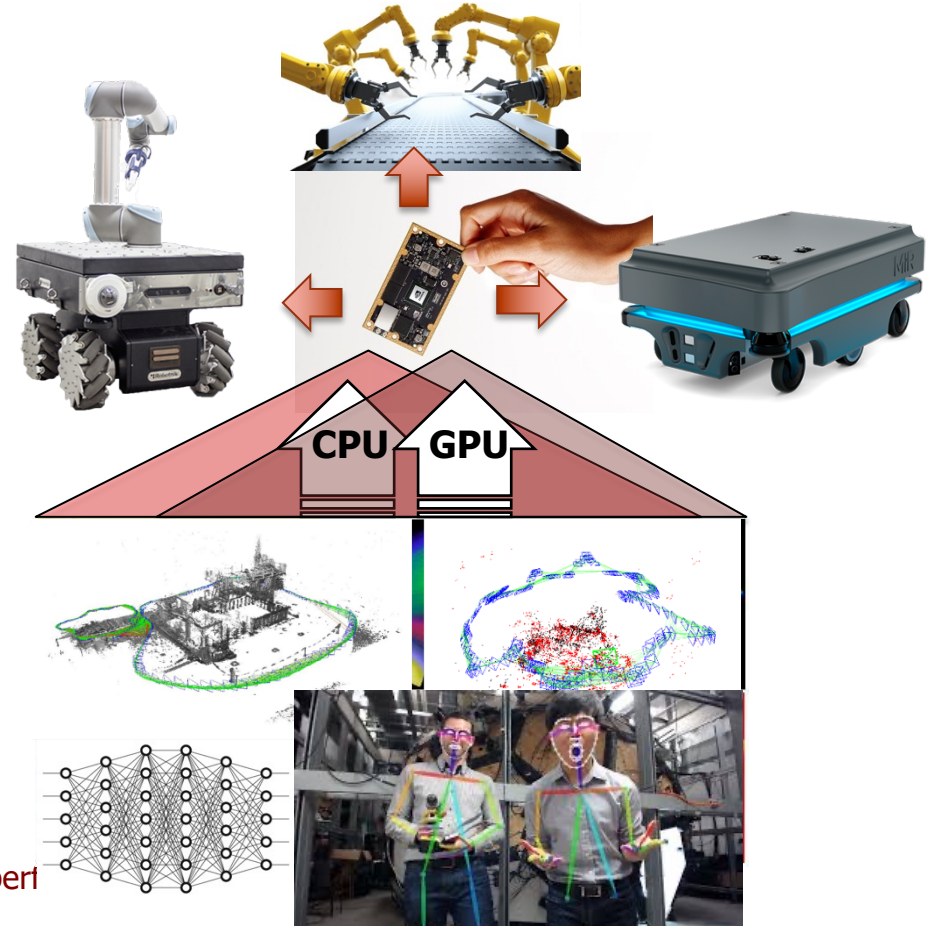


Embedded and IoT Systems Design - contributions

- System-level modeling of the entire cyber-physical production system (CPPS) **(16)**
 - entire production line
- Digital-twin modelling and simulation **(9)**
 - entire production line
- Embedded software design for embedded systems, PLCs and edge computers **(13)**
 - OPC-UA servers for each machinery
 - edge boards for data acquisition
- Integration of IoT and IIoT sensors into the edge and cloud **(14)**
 - sensors of rooms and machineries

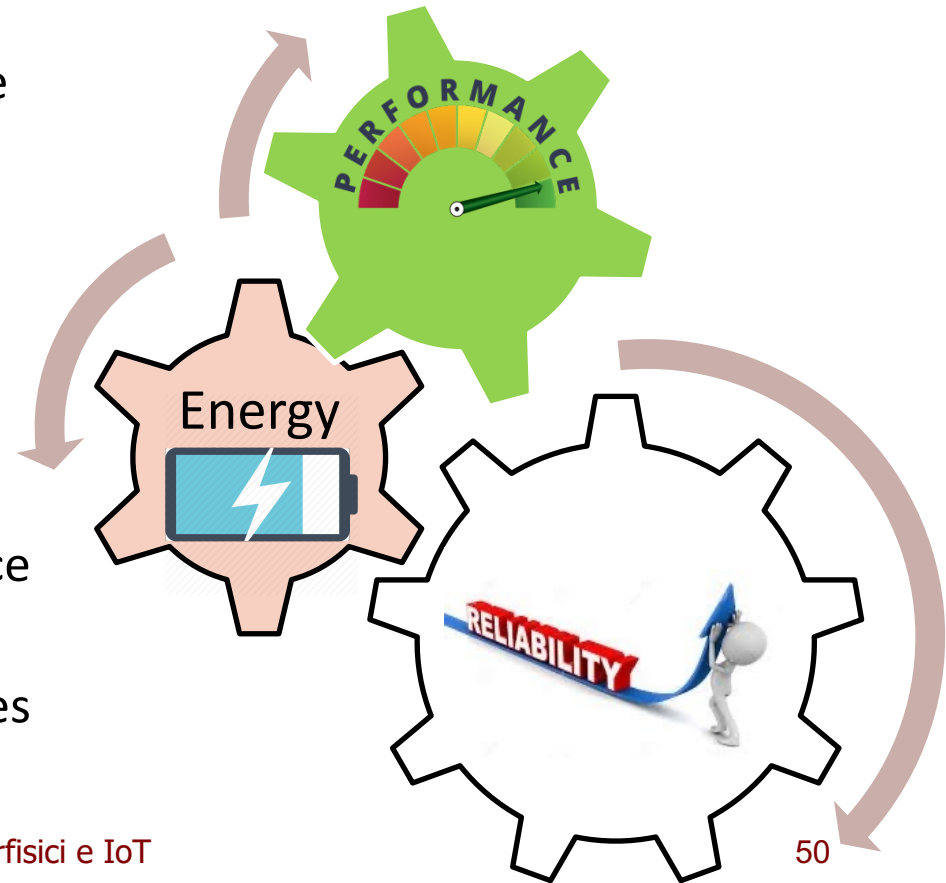
Advanced Computer Architectures - view

- Nicola Bombieri
- **Programming** of heterogeneous embedded devices for **edge computing**
 - Advanced SW programming targeting *performance* and *energy efficiency*
 - Multi-core CPU/Many-core (GPU) parallel programming
 - AI computing at the edge
 - Intelligent data processing at the edge



Advanced Computer Architectures - contributions

- Complex CPS+AI system performance and bottleneck analysis **(2)(11)**
 - From edge servers to edge devices
- Heterogeneous and parallel programming models for heterogeneous architectures **(13)**
 - From edge servers to edge devices
- Power/energy-aware SW performance tuning **(14)**
- Programming for real-time guarantees **(4)**



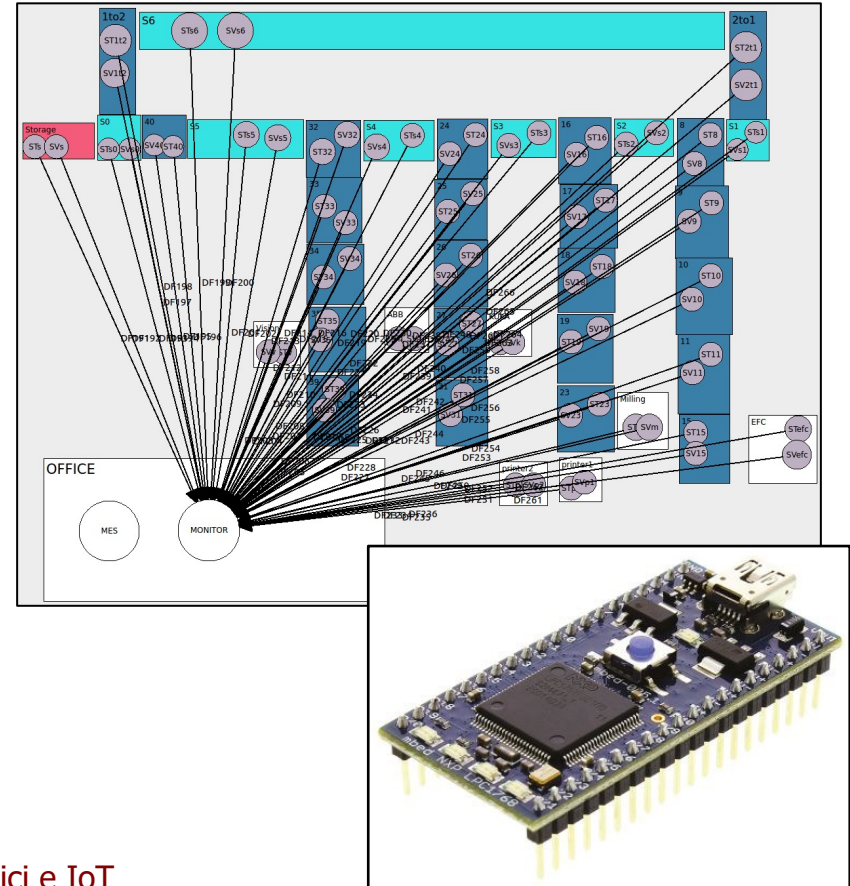
Networked Embedded & IoT Systems - view

- Methodologies and tools for modelling and design of networked embedded system for industrial applications
 - Network protocols for industry
 - Industrial IoT
 - Networked control systems

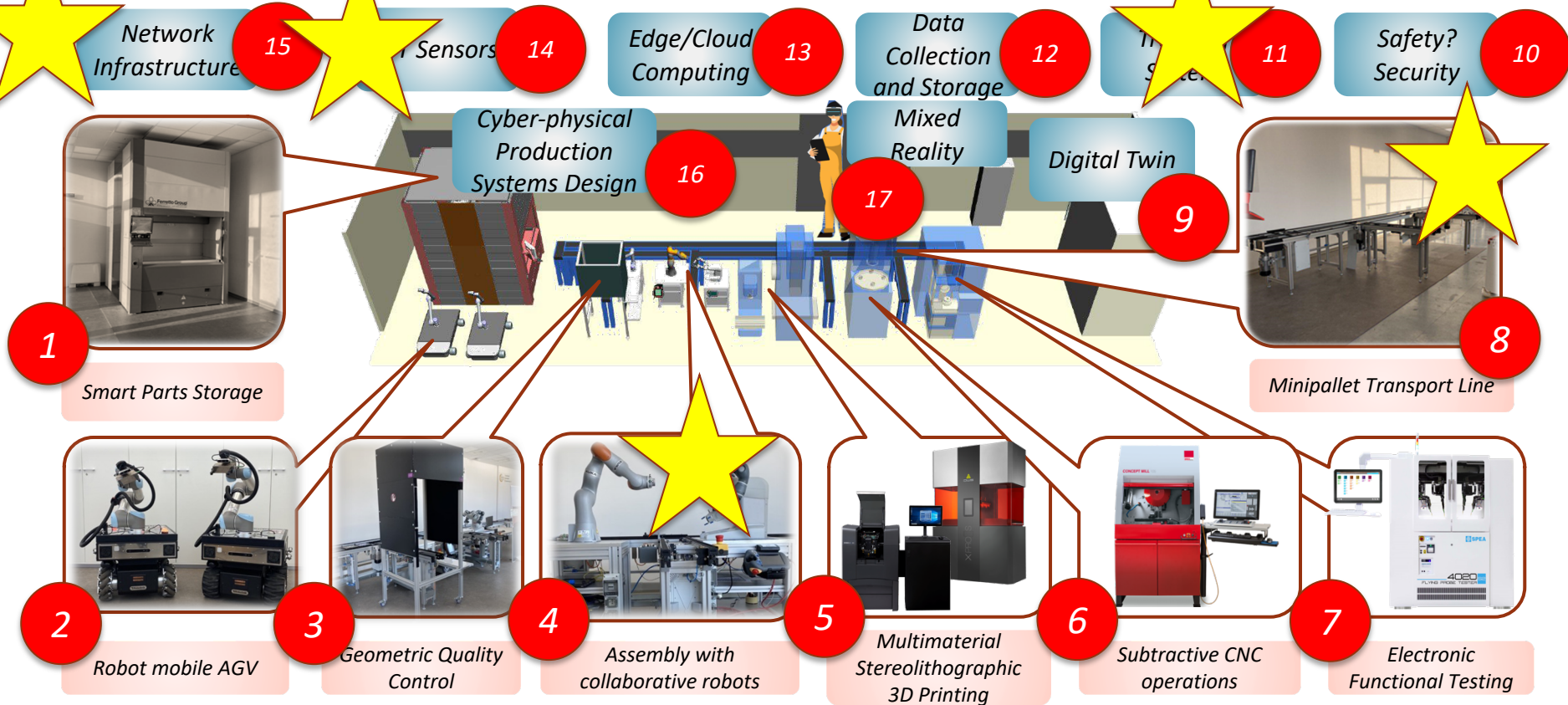


Networked Embedded & IoT Systems - contributions

- **Network synthesis**
 - Design of the communication infrastructure starting from constraints (e.g., delay) and goals (e.g., reliability, cost)
 - Network segregation of nodes for security enforcement with automatic generation of configuration files
- **Network simulation**
 - Assessment of the impact of OPC-UA flows over the overall communication infrastructure
- **Design of networked control applications in Lab ICE (joined lab with Embedded Operating Systems)**
 - Embedded node based on NXP LPC1768 with FreeRTOS

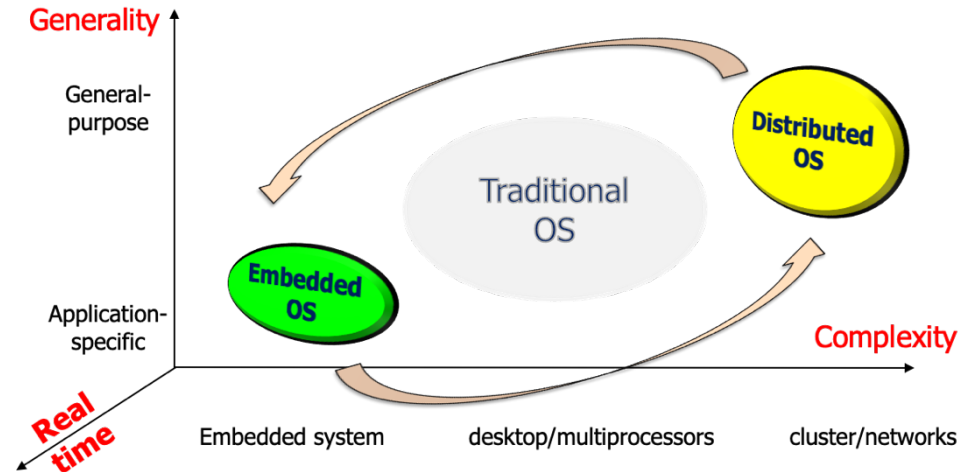


A completely reconf. production line



Embedded Operating Systems - view

- Graziano Pravadelli
- Advanced aspects of embedded operating systems
 - Design principles and constraints for OS in CPPSs and IIoT
 - Process and resource management in a distributed and real-time scenario



Embedded Operating Systems - contributions

- Embedded operating system principles in a real-time and distributed context **(13, 16)**
 - Real-time resource management
 - at the edge level
 - Distributed resource management
 - at the cloud level

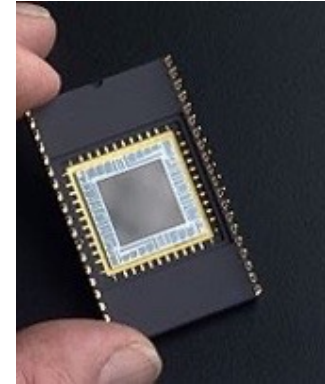
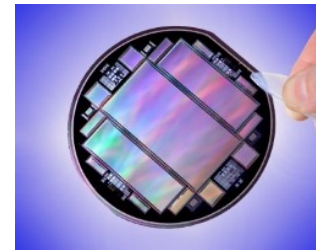
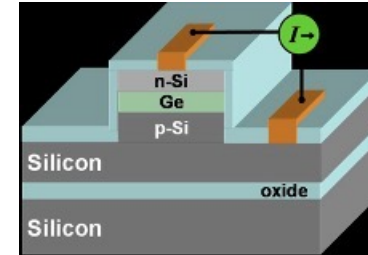
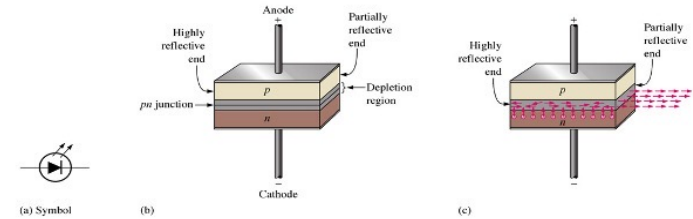
- Electronic Devices and Sensors – Romeo
- Systems Design Laboratory – Villa
- Systems Verification and Testing – Fummi/Pravadelli

ON CHOICE

Electronics Devices and Sensors

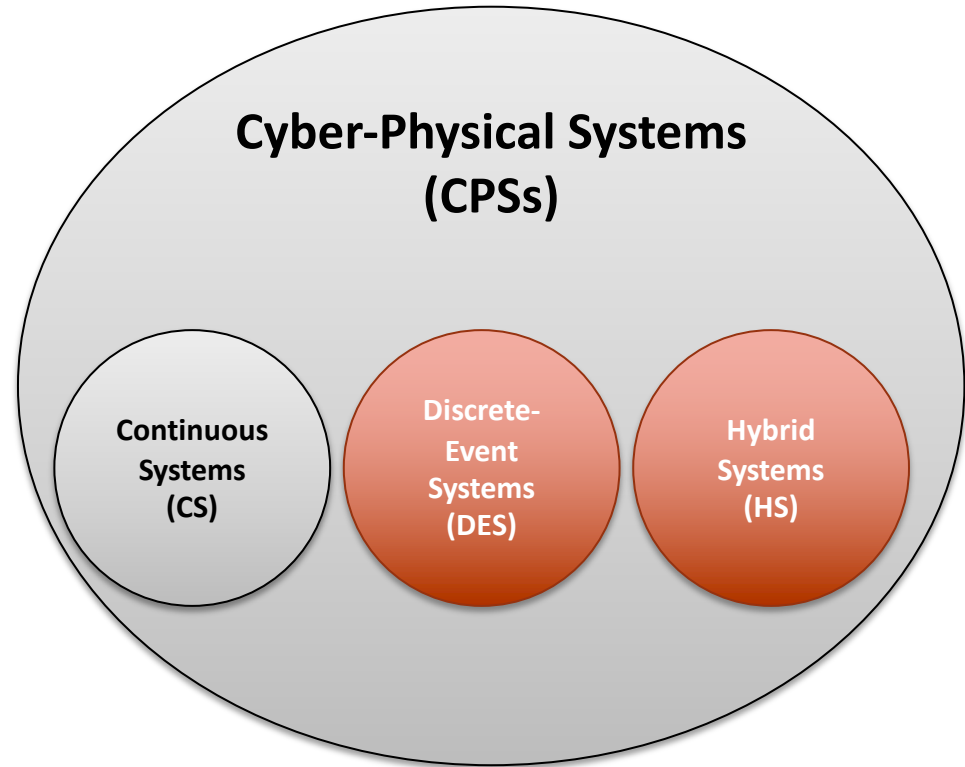
Alessandro Romeo

- Circuits
- Semiconductors
- Operation of p/n junctions
- Transistors
- Optical and electrical sensors



Systems Design Laboratory - view

- Tiziano Villa, Luca Geretti
- Formal methods for model-based design of engineering and manufacturing systems
 - Specification
 - Analysis
 - Synthesis of control
 - Planning/Scheduling



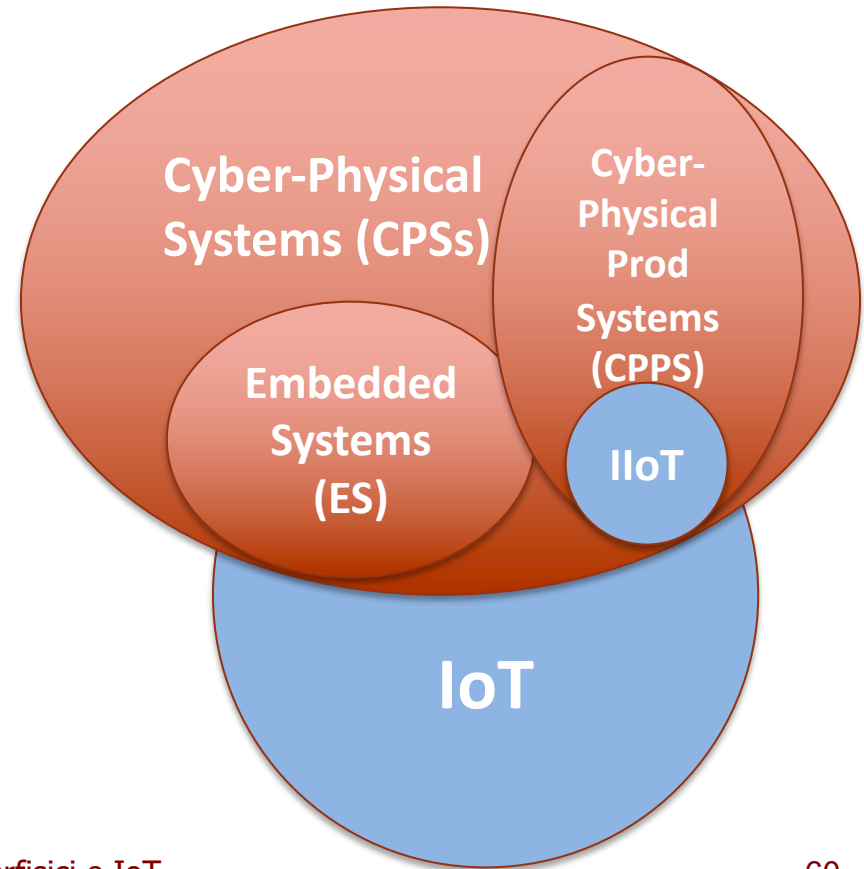
Systems Design Laboratory - contributions

This class leverages on the foundations introduced in Discrete and Hybrid Systems to introduce and employ state-of-the-art tools for the design of cyber-physical production systems **(16)**

- Modeling of a DES/HS using state-of-the-art languages and tools
- Focus on a detailed representation of the environment
- Verify safety properties of controlled and robotic systems, both offline and online
- Design of supervisory controllers of DES using libraries established in the industry
- Planning and scheduling of industrial production problems using a formal methodology

Systems Verification and Testing - view

- Franco Fummi / Graziano Pravadelli
- Methodologies and tools for verifying and testing CPPSs and IIoT
 - system verification models and languages
 - testing and functional safety
 - validation and verification
 - certification



Systems Verification and Testing - contributions

- System-level modeling for validation and verification of the entire cyber-physical production system (CPPS) **(16)**
 - entire production line
- Digital-twin simulation for verification **(9)**
 - entire production line
- Embedded software validation and testing from PLCs to edge computers **(13)**
 - Kubernetes-centric software validation and testing
- Functional testing of electronic boards **(7)**
 - test plan generation and application
 - testing machine control