



# UNICUSANO

Università degli Studi Niccolò Cusano - Telematica Roma

<b>Insegnamento</b>	Sistemi Embedded
<b>Livello e corso di studio</b>	Laurea Magistrale in Ingegneria Informatica (LM32)
<b>Settore scientifico disciplinare (SSD)</b>	ING-INF/05
<b>Anno di corso</b>	1
<b>Anno accademico</b>	2020/2021
<b>Numero totale di crediti</b>	12
<b>Propedeuticità</b>	-
<b>Docente</b>	Armando Piccardi Facoltà: Ingegneria Nickname: armando.piccardi Email: armando.piccardi@unicusano.it Orario di ricevimento: Consultare il calendario alla pagina seguente del nostro sito verificando gli orari di Videoconferenza <a href="https://www.unicusano.it/calendario-lezioni-in-presenza/calendario-area-ingegneristica">https://www.unicusano.it/calendario-lezioni-in-presenza/calendario-area-ingegneristica</a>
<b>Presentazione</b>	Il Corso di Sistemi Embedded ha lo scopo di far acquisire allo studente una buona conoscenza dei fondamenti scientifici e tecnologici delle architetture dei sistemi di elaborazione, le conoscenze e le abilità minime necessarie al progetto e alla programmazione degli elementi costruttivi degli elaboratori. Il Corso propone i fondamenti dei circuiti digitali e le tecniche di progettazione, a partire dalla conoscenza dell'organizzazione dell'architettura dei sistemi di elaborazione. Inoltre, obiettivo formativo del Corso è fornire allo studente una conoscenza nel dettaglio sulla struttura dei sistemi di elaborazione embedded, ed una conoscenza approfondita della programmazione di sistema per soluzioni embedded. Le Attività associate al Corso sviluppano le competenze necessarie a progettare architetture IoT e programmare sistemi embedded attraverso toolchain e simulatori.
<b>Obiettivi formativi</b>	Il Corso di Informatica ha i seguenti obiettivi formativi: <ol style="list-style-type: none"><li>1. Illustrare i fondamenti dei circuiti digitali e le tecniche di progettazione</li><li>2. Illustrare l'organizzazione dell'architettura dei sistemi di elaborazione</li><li>3. Illustrare la struttura dei sistemi di elaborazione embedded</li><li>4. Illustrare i principali strumenti di programmazione di sistema ed embedded</li></ol>

<b>Risultati di apprendimento attesi</b>	<p><b>Conoscenza e capacità di comprensione</b>  Lo studente al termine del Corso avrà conoscenza dei fondamenti scientifici e tecnologici delle architetture dei sistemi di elaborazione, delle caratteristiche dei diversi livelli e moduli che li compongono e le loro relazioni con gli strumenti di programmazione. Inoltre, lo studente conoscerà l'evoluzione storica e lo stato dell'arte dei sistemi di elaborazione embedded dal punto di vista teorico e strutturale. Lo studente verrà quindi edotto circa i principi di funzionamento delle principali architetture di tali sistemi e dei loro componenti. Inoltre, tramite le Etivity gli studenti acquisiranno una conoscenza approfondita della programmazione a basso livello e dei relativi strumenti di sviluppo.</p> <p><b>Applicazione delle conoscenze</b>  Lo studente sarà in grado di utilizzare le conoscenze acquisite attraverso l'analisi, la progettazione, la programmazione, e l'ottimizzazione di moduli e unità di controllo a diversi livelli dell'architettura degli elaboratori; sarà inoltre in grado di applicare le conoscenze acquisite prospettando possibili soluzioni a problemi di programmazione embedded, e saprà affrontarne l'implementazione. Le Etivity prevedono l'applicazione delle conoscenze teoriche a problemi pratici, come per esempio applicare la conoscenza del linguaggio assembly ed acquisire la capacità di usare tale linguaggio per scrivere programmi anche complessi.</p> <p><b>Capacità di trarre conclusioni</b>  Lo studente sarà in grado di valutare le esigenze tecniche che si presentano nella progettazione di circuiti (costo di progettazione, velocità di elaborazione) e di prendere decisioni metodologiche in merito alla progettazione dell'architettura di un calcolatore; sarà inoltre in grado di affrontare in autonomia l'analisi, la progettazione e l'implementazione di soluzioni per problemi di programmazione embedded e sarà in grado di valutarne la qualità in termini di semplicità, versatilità ed efficienza. Infine, lo studente sarà in grado di effettuare ricerche bibliografiche, di analizzare ed interpretare le fonti rilevanti, al fine di analizzare nuove architetture, e soluzioni embedded.</p> <p><b>Abilità comunicative</b>  Lo studente acquisirà la conoscenza dei formalismi e della terminologia usati per la descrizione di circuiti ed elementi architetturali dei calcolatori. Lo studente sarà in grado di descrivere e sostenere conversazioni su tematiche relative ai principi strutturali e di funzionamento dei calcolatori embedded, e di presentare analisi e soluzioni di problemi affrontabili con la programmazione embedded, adoperando una terminologia adeguata.</p> <p><b>Capacità di apprendere</b></p>
--	--

	<p>Lo studente al termine del Corso avrà conoscenza delle nozioni fondamentali necessarie per l'analisi e la progettazione di elaboratori. Questo gli consentirà di rapportare e confrontare le caratteristiche dell'organizzazione e della progettazione degli elaboratori. Lo studente sarà in grado di affrontare in maniera autonoma lo studio dei calcolatori embedded, con particolare riferimento a quanto disponibile in commercio e nella letteratura scientifica. Saprà affrontare problemi di programmazione embedded, individuando ed integrando librerie e soluzioni di terze parti. Sarà in grado di approfondire in autonomia la conoscenza di ulteriori strumenti di progettazione e di sviluppo.</p>
<b>Prerequisiti</b>	-
<b>Organizzazione dell'insegnamento</b>	<p>Il Corso è sviluppato attraverso le <b>lezioni preregistrate audio-video</b> che compongono, insieme a slide e dispense, i materiali di studio disponibili in piattaforma.</p> <p>Sono poi proposti dei <b>test di autovalutazione</b>, di tipo asincrono, che corredano le lezioni preregistrate e consentono agli studenti di accertare sia la comprensione, sia il grado di conoscenza acquisita dei contenuti di ognuna delle lezioni.</p> <p>La <b>didattica interattiva</b> è svolta nel forum della "classe virtuale" e comprende <b>9 Eivity</b> che applicano le conoscenze acquisite nelle lezioni di teoria alla progettazione di reti logiche, l'analisi di architetture di elaboratori e sistemi embedded, ed alla programmazione assembly e di firmware.</p> <p>In particolare, il Corso di Sistemi Embedded prevede 12 Crediti formativi. Il carico totale di studio per questo modulo di insegnamento è di circa 300 così suddivise in:</p> <ul style="list-style-type: none"> <li>• circa 215 ore per la visualizzazione e lo studio del materiale videoregistrato;</li> <li>• circa 75 ore di Didattica Interattiva per l'elaborazione e la consegna di 9 Eivity;</li> <li>• circa 10 ore di Didattica Interattiva per l'esecuzione dei test di autovalutazione.</li> </ul>
<b>Contenuti del corso</b>	<p><b>Modulo 1 – Reti logiche (impegno di 33 ore)</b> dove sono affrontati i seguenti argomenti: Nozioni introduttive e fondamentali dei sistemi di elaborazione. Struttura generale e livelli di astrazione dei sistemi di elaborazione. Compilazione e interpretazione. Rappresentazione binaria e algebra di Boole. Concetti di base dei circuiti logici combinatori. Definizione di reti combinatorie, loro formalizzazione e comportamento. Specifica di reti combinatorie. Procedimento di sintesi. Forme canoniche e semplificazione. Definizione di reti sequenziali, loro formalizzazione e comportamento. Classificazione e tipi di reti sequenziali. Esercitazione su logica combinatoria.</p> <p><b>Modulo 2 - Reti sequenziali, Architetture hardware (impegno di 26 ore)</b> dove sono affrontati i seguenti argomenti: Progettazione e sintesi di reti sequenziali. Reti sequenziali realizzate con componenti standard e loro analisi. Componente logico memoria. Unità di elaborazione: Modello Parte Controllo - Parte Operativa. Il procedimento di progettazione delle unità di elaborazione. Macchina di Von-Neumann. Modello di esecuzione FETCH-DECODE-EXECUTE. Introduzione alla microprogrammazione e implementazione dell'unità di controllo della CPU. Esercitazione su logica sequenziale.</p> <p><b>Eivity 1 (9 ore di carico di studio)</b> – Svolgimento guidato di esercizi sulla progettazione di reti combinatorie e sequenziali.</p> <p><b>Modulo 3 – Sistemi embedded (impegno di 14 ore)</b> dove sono affrontati i seguenti argomenti: Sensori e attuatori. Processori embedded. Architetture di memoria. Input e output. Multitasking. Scheduling. Bus.</p> <p><b>Eivity 2 (9 ore di carico di studio)</b> – Svolgimento guidato di esercizi sulla progettazione e l'analisi delle unità di elaborazione.</p> <p><b>Modulo 4 – Assembly ARM: introduzione (impegno di 24 ore)</b> dove sono affrontati i seguenti argomenti: Linguaggio macchina e linguaggio assembly ARM. Istruzioni di elaborazione dati. Flag di condizione. Salti. Costrutti di selezione. Cicli. Operazioni in memoria. Chiamate a sottoprogrammi. Gestione dello stack. Esercitazione su assembly.</p> <p><b>Eivity 3 (8 ore di carico di studio)</b> – Il livello della macchina assembler: assembleri, metodi di indirizzamento. Processori ARM: gestione della memoria, tipi di dato e istruzioni.</p> <p><b>Modulo 5 – Assembly ARM: aritmetica, estensioni (impegno di 24 ore)</b> dove sono affrontati i seguenti argomenti: Aritmetica intera e floating point. Istruzioni vettoriali. Estensioni multimediali. Esercitazione su assembly.</p> <p><b>Eivity 4 (10 ore di carico di studio)</b> – Laboratorio di programmazione assembly e debugging ARM.</p> <p><b>Modulo 6 – Architettura BCM2835/7 (impegno di 10,5 ore)</b> dove sono affrontati i seguenti argomenti: Organizzazione della memoria. Processo di boot. GPIO e Alternate Function. Timer hardware. Timer general-purpose. Interruzioni e controllore delle interruzioni. Tabella dei vettori di interruzione e routine di servizio (ISR). GPU.</p> <p><b>Eivity 5 (8 ore di carico di studio)</b> – Laboratorio virtuale basato su emulatori ed evaluation board remote.</p>

	<p><b>Modulo 7 – Architettura STM32-F446 (impegno di 10,5 ore)</b> dove sono affrontati i seguenti argomenti: Organizzazione della memoria. Clock. GPIO e Alternate Function. Controller I2C. Timer hardware. Timer general-purpose. Interruzioni e controllore delle interruzioni. Tabella dei vettori di interruzione e routine di servizio (ISR).</p> <p><b>Etivity 6 (8 ore di carico di studio)</b> – Laboratorio virtuale basato su emulatori ed evaluation board remote.</p> <p><b>Modulo 8 – Forth: linguaggio (impegno di 21,5 ore)</b> dove sono affrontati i seguenti argomenti: L’ambiente interattivo e l’interprete. Lo stack. Parole e loro definizione e compilazione. Notazioni di stack. Il dizionario. Interpretazione ed esecuzione. Operazioni aritmetiche con lo stack. Manipolazioni dello stack. Input e output. Costrutti di selezione. Cicli. Tipi numerici. Variabili, costanti e array. Esercitazione.</p> <p><b>Modulo 9 – Forth: interprete (impegno di 21,5 ore)</b> dove sono affrontati i seguenti argomenti: Implementazione dell’interprete e del dizionario. Execution token. Esecuzione vettoriale. Interprete interno ed esterno. Variabili di sistema. PAD. Text Input Buffer. Dizionario utente. Stack dei valori di ritorno. Variabili utente. Vocabolari. Esercitazione.</p> <p><b>Etivity 7 (10 ore di carico di studio) – Programmazione Forth.</b></p> <p><b>Modulo 10 – I/O, HAL (impegno di 26 ore)</b> dove sono affrontati i seguenti argomenti: Input and Output. General-purpose Input/Output. Bus paralleli. Interconnessioni seriali. UART e USART. Serial Peripheral Interconnect (SPI). Bus I2C. Programmazione di driver e Hardware Abstraction Layer. Esercitazione su programmazione GPIO.</p> <p><b>Etivity 8 (10 ore di carico di studio)</b> – Programmazione di driver e Hardware Abstraction Layer.</p> <p><b>Esercitazioni su compiti d’esame (4 lezioni di esercitazione per un impegno di 10 ore)</b></p> <p><b>Etivity 9 (6 ore di carico di studio)</b> – Simulazione esame.</p>
<p><b>Materiali di studio</b></p>	<p>· MATERIALI DIDATTICI A CURA DEL DOCENTE</p> <p>Il materiale didattico presente in piattaforma è suddiviso in 10 moduli. Essi ricoprono interamente il programma e ciascuno di essi contiene dispense, slide e videolezioni in cui il docente commenta le slide. Tale materiale contiene tutti gli elementi necessari per affrontare lo studio della materia.</p> <p>Testi consigliati:</p> <ul style="list-style-type: none"> <li>M. Vanneschi, Architettura degli Elaboratori, 2a Ed., Pisa University Press (2013)</li> <li>E. A. Lee and S. A. Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, 2nd Ed., MIT Press</li> <li>S. L. Harris, D. M. Harris, Digital Design and Computer Architecture: ARM Edition, Morgan Kaufmann</li> <li>L. Brodie, Starting Forth (on-line)</li> </ul>
<p><b>Modalità di verifica dell’apprendimento</b></p>	<p>L’esame consiste nello svolgimento di una <b>prova scritta</b> tendente ad accertare le capacità di analisi e rielaborazione dei concetti acquisiti e di una <b>serie di attività (e-tivity)</b> svolte durante il Corso nelle <b>classi virtuali</b>.</p> <p>I risultati di apprendimento attesi circa le conoscenze della materia e la capacità di applicarle sono valutate dalla prova scritta, mentre le abilità comunicative, la capacità di trarre conclusioni e la capacità di autoapprendimento sono valutate in itinere attraverso le e-tivity.</p>
<p><b>Criteri per l’assegnazione dell’elaborato finale</b></p>	<p>L’assegnazione dell’<b>elaborato finale</b> avverrà sulla base di un colloquio con il docente in cui lo studente manifesterà i propri specifici <b>interessi</b> in relazione a qualche argomento che intende approfondire; non esistono <b>preclusioni</b> alla richiesta di assegnazione della tesi e non è prevista una <b>media particolare</b> per poterla richiedere.</p>