



# UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria
ANNO ACCADEMICO OFFERTA	2021/2022
ANNO ACCADEMICO EROGAZIONE	2021/2022
CORSO DILAUREA MAGISTRALE	INGEGNERIA INFORMATICA
INSEGNAMENTO	EMBEDDED SYSTEMS
TIPO DI ATTIVITA'	B
AMBITO	50369-Ingegneria informatica
CODICE INSEGNAMENTO	21501
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	PERI DANIELE                      Ricercatore                      Univ. di PALERMO
ALTRI DOCENTI	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	96
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	54
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	1
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	<b>PERI DANIELE</b> Lunedì     15:00    16:00    Ricevimento in modalita a distanza sulla piattaforma MS Teams Mercoledì   15:00    16:00    Ricevimento in modalita a distanza sulla piattaforma MS Teams

<b>PREREQUISITI</b>	Fondamenti di programmazione, di circuiti logici e di architetture dei calcolatori. Conoscenza approfondita del linguaggio C.
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	<p><b>Conoscenza e capacita' di comprensione</b> Lo studente conoscerà l'evoluzione storica e lo stato dell'arte dei sistemi di elaborazione embedded dal punto di vista teorico e strutturale. Conoscerà quindi i principi di funzionamento delle principali architetture di tali sistemi e dei loro componenti. Avrà conoscenza approfondita della programmazione a basso livello e dei relativi strumenti di sviluppo.</p> <p><b>Capacita' di applicare conoscenza e comprensione</b> Lo studente sarà in grado di applicare le conoscenze acquisite prospettando possibili soluzioni a problemi di programmazione embedded, e saprà affrontarne l'implementazione.</p> <p><b>Autonomia di giudizio</b> Lo studente sarà 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.</p> <p><b>Abilità comunicative</b> Lo studente sarà in grado di esporre, efficacemente e con proprietà di linguaggio, i principi strutturali e di funzionamento dei calcolatori embedded. Saprà inoltre presentare analisi e soluzioni di problemi affrontabili con la programmazione embedded.</p> <p><b>Capacita' d'apprendimento</b> 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 e integrando soluzioni parziali già disponibili, sia formalizzate sia implementate. Sarà in grado di approfondire in autonomia la conoscenza di strumenti di progettazione e di sviluppo. Sarà inoltre in grado di integrare le conoscenze acquisite negli altri insegnamenti con quelle del corso.</p>
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	<p>La valutazione dell'apprendimento si basa sulla valutazione dei risultati attesi in accordo con i descrittori di Dublino. Il voto finale viene espresso in trentesimi e varia da 18/30 a 30/30 con lode.</p> <p>- Valutazione della conoscenza e capacita' di comprensione Per la verifica di questo obiettivo l'esame comprende una discussione orale sugli argomenti del programma. L'obiettivo contribuisce per il 25% al voto finale.</p> <p>- Valutazione della capacita' di applicare conoscenza e comprensione Per la verifica di questo obiettivo l'esame comprende lo sviluppo completo di una applicazione embedded, corredata da documentazione testuale che ne descriva scopo, struttura, funzionamento e modalità di utilizzo, a partire da specifiche testuali fornite dal docente. L'applicazione viene sviluppata dallo studente, singolarmente o in gruppo, al termine del corso e deve essere presentata funzionante al momento della discussione orale. L'obiettivo contribuisce per il 25% al voto finale.</p> <p>- Valutazione dell'autonomia di giudizio Nello sviluppo dell'applicazione embedded allo studente si richiede di effettuare delle scelte progettuali in autonomia. L'obiettivo viene quindi valutato discutendo nel dettaglio, durante l'esame orale, le scelte progettuali e implementative compiute nello sviluppo dell'applicazione assegnata. L'obiettivo contribuisce per il 20% al voto finale.</p> <p>- Valutazione delle abilità comunicative Per la verifica di questo obiettivo l'esame comprende la valutazione della documentazione dell'applicazione e la discussione dell'applicazione durante la prova orale. L'obiettivo contribuisce per il 10% al voto finale.</p> <p>- Valutazione della capacita' di apprendere Per la verifica di questo obiettivo l'esame comprende la discussione di alcuni argomenti avanzati introdotti a lezione e il cui approfondimento è lasciato agli studenti per la realizzazione dell'applicazione embedded. In particolare, l'obiettivo viene valutato discutendo le tecniche apprese autonomamente dallo studente e impiegate nella realizzazione dell'applicazione. L'obiettivo contribuisce per il 20% al voto finale.</p> <p>La formulazione delle prove fornisce una valutazione dei risultati attesi in relazione al voto finale come segue:</p> <p>- da 18/30 a 20/30: sufficiente conoscenza e capacita' di comprensione degli argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la</p>

	<p>risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere.</p> <p>- da 21/30 a 23/30: discreta conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere.</p> <p>- da 24/30 a 26/30: buona conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere.</p> <p>- da 27/30 a 30/30 e lode: eccellente conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere.</p>
<b>OBIETTIVI FORMATIVI</b>	Al termine del corso lo studente conoscerà i concetti avanzati necessari alla comprensione della struttura dei sistemi di elaborazione embedded. Avrà conoscenza approfondita della programmazione di sistema ed embedded.
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni ed esercitazioni in aula informatica.
<b>TESTI CONSIGLIATI</b>	<p>- E. A. Lee and S. A. Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, Second Edition, MIT Press, ISBN 978-0-262-53381-2, 2017, on-line (<a href="http://LeeSeshia.org">http://LeeSeshia.org</a>)</p> <p>- S. L. Harris, D. M. Harris, Digital Design and Computer Architecture: ARM Edition, Morgan Kaufmann</p> <p>- L. Brodie, "Starting Forth", on-line</p> <p>- A. S. Tanenbaum, T. Austin, "Structured computer organization. 6th ed.", Pearson</p> <p>- D. A. Patterson, J. L. Hennessy, "Computer Organization and Design", Morgan Kaufmann</p> <p>- L. Brodie, "Thinking Forth", on-line</p>

## PROGRAMMA

ORE	Lezioni
6	Architetture dei sistemi embedded. Sensori e attuatori. Processori embedded. Architetture di memoria. Input e output. Multitasking. Scheduling. Bus.
5	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.
4	Linguaggio macchina e linguaggio assembly (ARM). Aritmetica intera e floating point. Istruzioni vettoriali. Estensioni multimediali.
5	Architettura BCM2835/7. 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.
3	Architettura STM32-F446. 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).
2	Forth. 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.
2	Forth. Costrutti di selezione. Cicli. Tipi numerici. Variabili, costanti e array.
2	Forth. Implementazione dell'interprete e del dizionario. Execution token. Esecuzione vettoriale. Interprete interno ed esterno. Variabili di sistema. PAD. Text Input Buffer. Dizionario utente.
2	Forth. Stack dei valori di ritorno. Variabili utente. Vocabolari.
5	Input and Output. General-purpose Input/Output. Bus paralleli. Interconnessioni seriali. UART e USART. Serial Peripheral Interconnect (SPI). Bus I2C.
5	Programmazione di driver e Hardware Abstraction Layer.
ORE	Esercitazioni
4	Assembly ARM.
4	Programmazione Forth.
5	Programmazione di driver e Hardware Abstraction Layer.