

# Programmazione Didattica Preventiva

Docenti **Prof. DONATI Lorenzo; Prof.ssa ALOISI Mara**

Materia **Tecnologie e Progettazione di Sistemi Informatici e di Telecomunicazione (TPSIT)**

Classe **3F (ITT) – indirizzo Informatica e Telecomunicazioni (articolazione Telecomunicazioni)**

Anno Scolastico **2019/2020**

Ore **113h annuali (4h settimanali)**

**Metodi** Lezione frontale in interazione. Lezione di laboratorio. **Alcune lezioni potranno essere tenute interamente o parzialmente in lingua inglese.**

**Mezzi e materiali didattici** Dispense da parte del docente (anche in forma elettronica); software (con prevalenza di software open-source o freeware); strumentazione di laboratorio; documentazione tecnica in lingua inglese disponibile liberamente su Internet; Internet.

**Spazi** Aula. Laboratorio di Elettronica/Informatica.

**Verifiche** Test scritti a scelta multipla e/o con domande aperte, interrogazioni. Verifiche di laboratorio. Alcune verifiche potranno essere somministrate (integralmente o parzialmente) in lingua inglese.

**Obiettivi Minimi Comuni e Trasversali** Conoscere i concetti teorici alla base della disciplina e le problematiche ad essi associate. Conoscenza della terminologia tecnica in italiano ed in inglese. Saper leggere e capire documentazione tecnica di livello medio in inglese. Saper utilizzare Internet per cercare informazioni rilevanti ai fini della comprensione degli argomenti o al fine di risolvere un problema tecnico. Acquisire consapevolezza degli aspetti etici delle applicazioni legate alla disciplina.

**Collegamenti interdisciplinari** Matematica (algebra, funzioni), Informatica (uso di Internet, algebra di Boole, rappresentazione dei tipi di dato, programmazione), Telecomunicazioni (circuiti elettronici, teoria dei segnali), Inglese (microlingua). Sistemi e Reti (totalità del programma, vedi nota successiva).

**NOTA:** *Dato che il docente di materia (Lorenzo Donati) insegna nella medesima classe anche la materia “Sistemi e Reti”, per agevolare l’apprendimento degli studenti, si è deciso (di comune accordo con gli studenti stessi) di razionalizzare la programmazione dei due insegnamenti. Infatti i contenuti di tali materie presentano numerosissimi punti in comune. Pertanto si è deciso di inserire i medesimi moduli didattici nei piani di lavoro di entrambe le materie e di prevedere uno sviluppo del programma in parallelo. In questo modo gli studenti potranno concentrarsi su meno argomenti contemporaneamente, ma in modo più intensivo. Questo dovrebbe permettere una maggiore flessibilità di sviluppo del programma, andando incontro alle esigenze degli alunni e permettendo loro un recupero precoce di eventuali lacune.*

*Le verifiche di entrambe le materie verranno effettuate in modo coordinato, ottimizzando così ulteriormente i tempi.*

## *Moduli Didattici*

<i>Modulo</i>	<i>Contenuti</i>	<i>Obiettivi Minimi</i>	<i>Tempi (ore)</i>
<b>Algebra di Boole</b>	Operazioni logiche elementari (AND, OR, NOT, XOR). Tabelle di verità. Uguaglianze ed identità logiche. Proprietà dell'algebra booleana. Principio di dualità. Operazioni bit a bit.	Saper costruire la tabella di verità di un'espressione logica. Conoscere le proprietà dell'algebra di Boole e saperle applicare alla manipolazione di espressioni logiche. Saper verificare un'identità logica. Saper effettuare calcoli con operazioni bit a bit tra parole binarie di lunghezza arbitraria.	15
<b>Reti Logiche</b>	Porte logiche (AND, NAND, OR, NOR, NOT, BUFFER, XOR, XNOR), e circuiti logici. Reti combinatorie e forme canoniche. Reti combinatorie standard (multiplexer, demultiplexer, decoder, etc.). Diagrammi di temporizzazione. Tempo di propagazione. Reti sequenziali sincrone ed asincrone. Reti sequenziali standard (latch, flip-flop, contatori, registri, etc.).	Conoscere le porte logiche, le loro leggi di commutazione e le loro tabelle di verità. Saper costruire il circuito logico a partire dalla sua legge di commutazione e viceversa. Saper determinare la forma canonica SP partendo dalla tabella di verità. Saper costruire un diagramma di temporizzazione, anche tenendo conto di eventuali ritardi di propagazione, e saperlo interpretare. Conoscere e saper utilizzare le reti standard (combinatorie e sequenziali) nelle applicazioni più comuni.	15
<b>Elettronica Digitale</b>	Aspetti implementativi delle reti logiche in tecnologia elettronica e basi della tecnologia CMOS. Corrispondenza tra stati logici e livelli logici. Famiglie logiche integrate CMOS. Porte di trasmissione CMOS. Ingressi con trigger di Schmitt. Uscite 3-state, open-drain e open-collector. Bus bidirezionali. MUX/DEMUX analogici. Circuiti di generazione del clock.	Conoscere le principali caratteristiche e limitazioni della tecnologia CMOS. Saper leggere la documentazione tecnica inglese dei dispositivi digitali. Conoscere il funzionamento e le applicazioni dei principali dispositivi speciali disponibili grazie alla tecnologia CMOS (ingressi a trigger di Schmitt; porte di trasmissione; uscite 3-state, open-drain e open-collector; MUX/DEMUX analogici). Conoscere la struttura di alcuni tipici circuiti di generazione del clock.	10
<b>Architettura dei Sistemi di Elaborazione Digitale</b>	Componenti di un sistema: microprocessori, DSP, FPGA, microcontrollori, memorie, periferiche, porte ed interfacce di comunicazione. Struttura di un microprocessore e linguaggio macchina. Architettura di Von Neumann e architettura di Harvard. Meccanismo delle Interruzioni. Meccanismo di accesso diretto alla memoria (DMA).	Conoscere i principali dispositivi digitali programmabili, le diverse tecnologie di memorizzazione elettronica e la struttura di un microprocessore. Conoscere il funzionamento dei meccanismi di interruzione e di accesso diretto alla memoria.	20
<b>Programmazione Assembly</b>	Corrispondenza tra linguaggio macchina e linguaggio Assembly. Programma assemblatore. Linguaggio assembly del microcontrollore ATmega328P.	Conoscere i concetti fondamentali legati alla struttura del linguaggio macchina e la corrispondenza con l'Assembly. Saper capire semplici frammenti di codice Assembly della MCU ATmega328P.	10
<b>Microcontrollori</b>	Struttura hardware di un microcontrollore (MCU). Programmazione C del microcontrollore ATmega328P mediante piattaforma ed ambiente di sviluppo Arduino. Interfacciamento di un microcontrollore con dispositivi esterni per mezzo di linee di IO generico (GPIO). Utilizzo dei timer di una MCU per la generazione di segnali digitali e per la gestione di eventi.	Conoscere la struttura di un microcontrollore e le sue periferiche incorporate. Saper programmare il microcontrollore della piattaforma Arduino in linguaggio C/C++. Conoscere e saper applicare le tecniche di programmazione generali di un microcontrollore. Saper interfacciare la scheda Arduino con dispositivi esterni mediante linee GPIO. Saper scrivere semplici programmi C per la scheda Arduino UNO (senza usare le librerie di Arduino).	23
<b>Sistemi Operativi</b>	Struttura e funzioni di un sistema operativo.	Conoscere la struttura e le funzioni di un sistema operativo.	20