

ISTITUTO TECNICO INDUSTRIALE STATALE "MAX PLANCK"
VIA FRANCHINI, 1 31020 - LANCENIGO DI VILLORBA (TV)



ORGANISMO DI FORMAZIONE ACCREDITATO PRESSO LA REGIONE VENETO: COD. N. 218

PROGRAMMAZIONE DI DIPARTIMENTO

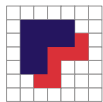
DIPARTIMENTO DI ELETTRONICA ED ELETTROTECNICA

Classe: TERZA

Articolazione: AUTOMAZIONE

Materia: Sistemi Automatici

A.S.: 2024-2025



Programmazione

La disciplina prevede 6 ore di lezione settimanali di cui 3 in compresenza (approvato in data 14/5/2018 dal Collegio dei Docenti (C.d.D.).

La programmazione di SISTEMI è stata strutturata in base alle indicazioni delle Linee Guida Ministeriali e alle indicazioni e osservazioni emerse in sede di riunione di dipartimento di materia.

Come si evince dalle linee guida ministeriali, il corso di Sistemi automatici è finalizzato a far conseguire i seguenti risultati di apprendimento relativi al profilo educativo, culturale e professionale:

- utilizzare, in contesti di ricerca applicata, procedure e tecniche per trovare soluzioni innovative e migliorative, in relazione ai campi di propria competenza;
- cogliere l'importanza dell'orientamento al risultato, del lavoro per obiettivi e della necessità di assumere responsabilità nel rispetto dell'etica e della deontologia professionale;
- riconoscere gli aspetti di efficacia, efficienza e qualità nella propria vita lavorativa;
- saper interpretare il proprio autonomo ruolo nel lavoro di gruppo;
- essere consapevole del lavoro sociale della propria attività, partecipando attivamente alla vita civile e culturale a livello locale, nazionale e comunitario;
- riconoscere e applicare i principi dell'organizzazione, della gestione e del controllo dei diversi processi produttivi;
- analizzare criticamente il contributo apportato dalla scienza e dalla tecnologia allo sviluppo dei saperi e al cambiamento delle condizioni di vita;
- riconoscere le implicazioni etiche, sociali, scientifiche, produttive, economiche e ambientali dell'innovazione tecnologica e delle sue applicazioni industriali.

Finalità educative e didattiche trasversali.

Si cercherà di sviluppare e potenziare le competenze cognitive trasversali di cui si è già sperimentato da vari anni il monitoraggio e la valutazione al biennio. Tra le competenze cognitive su cui concentrare l'attenzione si sono scelti i punti seguenti:

imparare ad imparare

Essere in grado di organizzare il proprio apprendimento e di rispettare i tempi di consegna.

Saper utilizzare la lingua scritta per prendere appunti in modo abbastanza preciso e saper commentare i passaggi nei compiti

comunicare efficacemente

Saper pianificare ed organizzare l'esposizione orale tenendo conto del destinatario della comunicazione

Verifiche e valutazioni

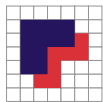
Le verifiche per il voto orale potranno essere svolte in forma orale o scritta (mediante tipologie quali prove strutturate, semi-strutturate, questionari, domande a risposta chiusa o aperta ecc. come previsto dal P.O.F. e/o dal documento di dipartimento.

Eventuali verifiche di recupero (individuali o a piccoli gruppi) potranno svolgersi in forma scritta o in forma orale per le diverse le tipologie di voto.

Oltre alle verifiche previste potranno essere effettuate semplici domande dal posto e controllo dei compiti assegnati per casa che saranno indicate come valutazioni formative e di cui si terrà conto complessivamente per l'arrotondamento del voto finale del quadrimestre e dell'anno scolastico.

Le prove di laboratorio consisteranno nella realizzazione del software proposto, nel suo collaudo e relativa correzione, ed eventualmente nella produzione della relazione cartacea sulla conduzione e risultati della prova pratica in laboratorio.

Ciascuna prova di laboratorio sarà svolta singolarmente o in gruppo (costituito generalmente da non più di 2 allievi) in funzione del tipo di prova, di necessità didattiche e del numero di postazioni disponibili.



Saranno svolti in caso di necessità ripassi e recuperi curricolari oltre a quanto deciso in sede di C.d.C. e redatto in appositi verbali a cui si rimanda.

Come deciso durante le riunioni del Dipartimento di materia e redatto negli appositi verbali a cui si rimanda, il numero minimo di verifiche è :

- ☐ due per il primo quadrimestre;
- ☐ tre nel secondo quadrimestre.

Le parti facoltative di approfondimento per le eccellenze verteranno su argomenti trattati durante l'anno scolastico e saranno concordate con gli studenti per tener conto delle loro inclinazioni ed interessi. La valutazione sarà effettuata sulla base di una interrogazione orale e/o una produzione pratica e/o una tesina.

Per quanto concerne la valutazione, oltre a rimandare a quanto stabilito dal Collegio dei Docenti, dal P.O.F. e dal Consiglio di Classe in apposite riunioni e redatto nei relativi verbali, si fa riferimento alla tabella sotto riportata. Tale tabella di valutazione sarà usata come riferimento per redigere le griglie di valutazione per le prove scritte (allegate al testo di ogni verifica), per le prove orali e per le prove di laboratorio.

I criteri di valutazione utilizzati, le metodologie, gli strumenti, la tabella di riferimento per la valutazione delle prove di verifica, i metodi e le forme di recupero ecc. sono chiariti agli studenti ad inizio anno scolastico (e durante tutto l'anno scolastico qualora se ne manifestasse la necessità), nonché consegnate per le prove scritte e pratiche.

Le prove di verifica pratiche non consegnate in alcuna loro parte nei tempi previsti sarà valutata, in assenza di comprovate e valide motivazioni che abbiano impedito il loro svolgimento, con il minimo dei voti della tabella di valutazione del profitto.

Dopo ogni verifica orale lo studente sarà invitato ad autovalutarsi.

Tempi

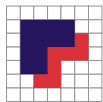
I tempi indicati nella programmazione si riferiscono all'espletamento dell'intera unità e comprendono pertanto l'accertamento e l'eventuale ripasso dei prerequisiti, le lezioni, il ripasso curricolare, le verifiche ed eventuali verifiche di recupero.

I tempi richiesti per lo svolgimento del lavoro domestico assegnato rientra nei tempi stabiliti dal consiglio di classe e dalle riunioni di dipartimento di materia e riportati nei relativi verbali a cui si rimanda.

Note sulla programmazione

- Per necessità didattiche la successione degli argomenti, delle unità didattiche e/o dei singoli contenuti adottata nella programmazione non è da ritenersi corrispondente all'ordine con cui saranno svolti a lezione pur rispettandone le propedeuticità.
- La durata effettiva delle spiegazioni e delle esercitazioni dipenderanno da come la classe affronta e risponde agli stimoli proposti dall'insegnante. Il numero di ore di verifica o ripasso svolte al singolo studente o ad un piccolo gruppo, che potranno essere svolte nelle ore di compresenza, non sarà scorporato. Sarà indicato solo nel caso il ripasso riguardi l'intera classe.
- Verifiche (es. orali) ed attività di recupero-ripasso sono da intendersi come un'attività strettamente legata in quanto la spiegazione dell'allievo interrogato e le conseguenti conferme e/o eventuali correzioni possono costituire un momento di ripasso per il singolo allievo e per la classe. Inoltre durante il ripasso possono svolgersi domande di tipo formativo come indicato precedentemente.
- Sarà seguito, quando possibile, il testo in adozione, saranno fornite dispense, quesiti, problemi per il ripasso della materia e si farà più volte appello alla cura degli appunti di lezione in modo da agevolare lo studio domestico. In molte parti della programmazione si fa esplicito riferimento al titolo degli argomenti presenti nel testo.
- L'attività di laboratorio, oltre a permettere allo studente di prendere confidenza con gli strumenti messi a disposizione dalla scuola, con i supporti informatici di elaborazione e simulazione, sarà utilizzata per approfondire e consolidare, attraverso attività pratiche, esercitazioni, simulazioni ecc., le parti trattate in classe. Pertanto tali attività seguiranno di norma, nei contenuti, nelle esperienze e nelle eventuali verifiche, l'attività teorica. Il tipo ed il numero di esperienze di laboratorio è molto variabile, in quanto dipendente dalle contingenze didattiche; tuttavia verranno effettuate, in media, almeno due prove pratiche a periodo.

Competenze generali comuni a tutte le unità



Le seguenti competenze generali si intendono riferite a tutti gli argomenti e unità della programmazione e quindi non saranno successivamente ripetute:

Saper

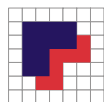
- Enunciare definizioni, teoremi, principi e saperli spiegare ed applicare nei diversi contesti.
- Descrivere e discutere le parti teoriche trattate.
- Riconoscere, interpretare ed utilizzare il linguaggio e la simbologia specifica della materia.
- Svolgere semplici calcoli e passaggi matematici, rappresentare mediante testo, grafici, diagrammi, tabelle ecc. gli elementi studiati, saperli interpretare e saper spiegare le relazioni reciproche tra le parti e/o gli elementi trattati.
- Analizzare nei minimi dettagli un problema anche complesso e tradurlo, mediante il metodo TOP-DOWN e/o BOTTOM-UP strutturato, in una procedura codificata (flowchart, pseudocodice e programma nello specifico linguaggio (es. C/C++/LabView/mikroc/C per Arduino) e viceversa descrivere l'algoritmo implementato da un programma in linguaggio C, o altra forma codificata, che impieghi gli elementi studiati.
- Interpretare correttamente, valutare la bontà dell'algoritmo (dato in una delle forme codificate) contenente i diversi elementi studiati.
- Scrivere un algoritmo dato in pseudocodice o flowchart che impieghi gli elementi studiati nel corrispondente programma in linguaggio in forma codificata e viceversa.
- Scegliere correttamente gli elementi studiati più opportuni per risolvere un determinato problema.
- Scrivere un programma nello specifico linguaggio (o altra forma codificata) senza errori sintattici e logici contenente gli elementi studiati e, se presenti, saperli individuare e correggere.
- Scrivere, compilare, aggiungere le librerie e produrre un file eseguibile di un programma in forma codificata contenente gli elementi studiati per mezzo dell'elaboratore, saperne effettuare il debug e saper utilizzare le principali funzioni dell'ambiente.
- Svolgere semplici calcoli e passaggi matematici, estrarre parametri caratteristici, tracciare ed interpretare grafici, schemi, diagrammi e schemi a blocchi, andamenti temporali, scrivere le relazioni analitiche ecc. relativamente alle parti trattate.
- Utilizzare correttamente le grandezze studiate e le relative unità di misura.
- Saper consultare i fogli tecnici (anche in lingua inglese) degli elementi studiati e saper ricavare le informazioni necessarie.

(relativamente all'attività di laboratorio, oltre a quanto sopra)

Saper

- Descrivere e discutere il proprio lavoro.
- Utilizzare componenti, dispositivi e strumentazione del laboratorio per realizzare quanto richiesto.
- Scrivere, collaudare, correggere commentare e discutere il software.
- Scrivere, compilare, gestire le librerie, effettuare il link e produrre un file eseguibile di un programma in forma codificata contenente gli elementi studiati per mezzo dell'elaboratore, saperne effettuare il debug (individuazione e correzione) e saper utilizzare le principali funzioni e strumenti dell'ambiente. Saper programmare nello specifico linguaggio in forma codificata usando la programmazione TOP-DOWN strutturata.
- Effettuare correttamente una ricerca mediante l'help dell'ambiente ed i manuali.
- Impostare le opzioni e proprietà dell'ambiente di sviluppo.
- Ricercare informazioni relative ad elementi di programmazione, algoritmi di risoluzione o quant'altro inerente la trattazione o l'esperienza da condurre, dalle fonti più comuni disponibili (libro di testo, manuali, guida in linea, CD, internet ecc.) e saperli consultare (anche se in lingua inglese).
- Utilizzare i diversi pacchetti applicativi per la stesura di una relazione tecnica quando richiesta (scrittura, disegno, calcolo ecc.).
- Produrre la documentazione che accompagna l'esperienza svolta in classe (descrizione ed analisi del problema, disegni, calcoli, progetto, presentazione dei dati reali e/o simulati, valutazione dei risultati ottenuti, possibili miglioramenti dell'esperienza svolta).

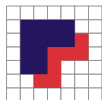
L'attività di laboratorio è strettamente connessa ai contenuti teorici dell'argomento corrispondente. **Si rimanda pertanto alle relative competenze.**



Descrittori dei voti delle prove di verifica

La tabella che segue riporta i descrittori dei voti del profitto generici. Le griglie di valutazione delle verifiche saranno formulate sulla base di queste indicazioni fornite dal P.T.O.F.

Voto	Conoscenza	Abilità	Competenze
10	Completa, precisa ed approfondita	<u>Esposizione</u> organica e originale. <u>Linguaggio</u> efficace e specifico. <u>Comprensione e risoluzione</u> logico-operative ottime di procedure tecniche disciplinari.	<u>Analisi, sintesi e rielaborazione</u> rigorosa, critica e personale. <u>Osservazione e interpretazione</u> precise e personali. <u>Uso</u> autonomo di procedure tecniche disciplinari in vari contesti. <u>Uso</u> autonomo e pertinente di strategie per la soluzione di problemi e processi.
9	Completa e precisa	<u>Esposizione</u> organica, completa e precisa. <u>Linguaggio</u> efficace e specifico. <u>Comprensione e risoluzione</u> logico-operative ottime di procedure tecniche disciplinari.	<u>Analisi, sintesi e rielaborazione</u> rigorosa e critica; <u>Osservazione e interpretazione</u> precise e personali. <u>Uso</u> autonomo di procedure tecniche disciplinari in vari contesti. <u>Uso</u> autonomo e pertinente di strategie per la soluzione di problemi e processi.
8	Completa	<u>Esposizione</u> completa. <u>Linguaggio</u> corretto e appropriato. <u>Comprensione e risoluzione</u> logico-operative appropriate di procedure tecniche disciplinari.	<u>Analisi, sintesi e rielaborazione</u> appropriati. <u>Osservazione e interpretazione</u> corrette e organiche. <u>Uso</u> corretto di procedure tecniche e simbologie disciplinari. <u>Uso</u> corretto di strategie per la soluzione di problemi e processi.
7	Abbastanza Completa e sostanzialmente sicura	<u>Esposizione</u> chiara e ordinata. <u>Linguaggio</u> adeguato ma non sempre specifico. <u>Comprensione e risoluzione</u> logico-operative adeguate di procedure tecniche disciplinari.	<u>Analisi, sintesi e rielaborazione</u> abbastanza autonome e precise. <u>Osservazione e interpretazione</u> non sempre puntuali di procedure e tecniche disciplinari. <u>Uso</u> parziale di strategie per la soluzione di problemi e processi.
6	Essenziale degli elementi principali della disciplina	<u>Esposizione</u> chiara e semplice. <u>Linguaggio</u> non sempre corretto. <u>Comprensione e risoluzione</u> logico-operative sufficienti di procedure tecniche disciplinari.	<u>Analisi, sintesi e rielaborazione</u> parziali con spunti autonomi. <u>Osservazione e interpretazione</u> sufficienti delle procedure tecniche e simbologie disciplinari. <u>Uso</u> complessivamente sufficiente di strategie per la soluzione di problemi e processi.
5	Superficiale, in presenza di errori	<u>Esposizione</u> generica e stentata. <u>Linguaggio</u> impreciso. <u>Comprensione e risoluzione</u> logico-operative parziali di procedure tecniche disciplinari.	<u>Analisi, sintesi</u> solo guidate. Effettua collegamenti solo parziali. <u>Osservazione e interpretazione</u> generiche di procedure tecniche e simbologie disciplinari. <u>Uso</u> impreciso di strategie per la soluzione di problemi e processi.
4	Frammentaria con errori rilevanti	<u>Esposizione</u> incerta e disorganica. <u>Linguaggio</u> approssimativo e improprio. <u>Comprensione e risoluzione</u> logico-operative scarse di procedure tecniche disciplinari.	<u>Analisi, sintesi</u> parziali e solo guidate. <u>Osservazione e interpretazione</u> lacunose e imprecise di procedure tecniche e simbologie disciplinari. <u>Uso</u> lacunoso e impreciso di strategie per la soluzione di problemi e processi.
3	Frammentaria e lacunosa degli elementi con errori gravi e diffusi	<u>Esposizione</u> stentata, confusa e disorganica. <u>Linguaggio</u> scorretto. <u>Comprensione e risoluzione</u> logico-operative stentate e scorrette di procedure tecniche disciplinari.	<u>Uso</u> molto lacunoso o assente di strategie per la soluzione di problemi e processi.
2	Quasi completamente errata	<u>Esposizione</u> confusa. <u>Linguaggio</u> approssimativo. <u>Comprensione e risoluzione</u> logico-operative assenti di procedure tecniche disciplinari.	Quasi assente
1	Completamente errata	<u>Esposizione</u> decisamente confusa. <u>Linguaggio</u> decisamente approssimativo. <u>Comprensione e risoluzione</u> logico-operative assenti di procedure tecniche disciplinari.	Assente



Finalità ed obiettivi specifici:

Comprendere adeguatamente il funzionamento di un PC
Saper usare con adeguata abilità il pc al fine di risolvere problemi con metodi algoritmici
Saper creare, date le specifiche, programmi in C/CPP, C per Arduino ed per Labview
Saper progettare semplici circuiti combinatori e sequenziali

Contenuti irrinunciabili

Codifiche, cambiamenti di base, algoritmi, programmazione in CPP, C per Arduino e Labview, componenti con memoria, analisi e sintesi di circuiti logici combinatori e sequenziali.

Conoscenze e abilità minime per l'accesso alla classe successiva

Saper sviluppare in CPP, C per Arduino e in Labview semplici programmi, comprendere il metodo di minimizzazione delle funzioni logiche e saperlo applicare, saper eseguire la sintesi di semplici circuiti combinatori (es. contatori)

Testo adottato

Mario Capuzzimati – “Sistemi Automatici” per le articolazioni “Automazione” di Elettronica Elettrotecnica – volume 1 - Tramontana – ISBN9788823374959

Testi consigliati

I libri in adozione nella stessa classe relativi alle materie di indirizzo (Elettronica ed Elettrotecnica, T.P.S.E.E.) utilizzati per rimandi, contenuti ed approfondimenti interdisciplinari, ricerche personali ecc.
- E&E a colori - Cuniberti, De Lucchi, Bobbio, Sammarco – Ed. DeA Scuola Petrini
G. Portaluri, E. Bove – “Tecnologie e progettazione di sistemi elettrici ed elettronici” – vol. 1 — Ed. Tramontana – edizione mista.
Sono compresi i contenuti digitali dei testi che gli allievi devono recuperare autonomamente.

Schema riassuntivo con le tempistiche dei moduli:

Modulo	Argomento	Ore previste (totale 190..)
1	Circuiti combinatori	25
2	Circuiti con memoria (sequenziali)	35
3	Elementi di informatica	25
4	Programmazione in C/C++	45
5	Programmazione con Arduino	30
6	Programmazione con LabView	30

Argomento	1	Circuiti combinatori					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 Richiami su elementi di Algebra Booleana	Nozioni elementari di algebra e di logica. I codici su base binaria Matematica del biennio	La definizione di variabile logica, di circuito combinatorio, funzione logica e tabella di verità. Le funzioni logiche primarie e le relative porte. Altre funzioni logiche ed i gruppi universali. Teorema di De Morgan.	Saper: <ul style="list-style-type: none"> Comprendere il comportamento logico di un circuito fornito in una delle forme studiate (tab. di verità, funz. logica, circuito combinatorio) e tradurlo in una qualsiasi altra. Scrivere la corretta tabella di verità di un circuito combinatorio che risolve un semplice problema assegnato. Verificare l'uguaglianza tra funzioni logiche. Utilizzare la logica affermata e negata. Analizzare, classificare e valutare un circuito logico. Tracciare e interpretare il diagramma temporale di uno o più segnali relativi ad un componente o ad un circuito combinatorio 	LF LI LG LM EC LT MA DI AL	VO	3	ST EX

U2 Funzione logica, forma canonica e metodi di minimizzazione in forma canonica.	Argomenti ed unità precedenti e relativi prerequisiti	<p>Forme canoniche. Le mappe di Karnaugh, il metodo di minimizzazione e di risoluzione a porte dello stesso tipo. Le condizioni di indifferenza. Criteri e metodo di ottimizzazione Livelli logici, logica positiva e negativa. Progetto dei circuiti combinatori. Il diagramma temporale di un componente o circuito combinatorio. Comprendere il problema delle alee statiche</p>	<p>Saper:</p> <ul style="list-style-type: none"> • Tracciare le mappe di Karnaugh in base alla funzione logica, tabella di verità e/o diagramma temporale e viceversa. • Minimizzare le funzioni logiche • Progettare il circuito combinatorio minimo e/o con porte di un solo tipo che realizzi una assegnata funzione logica esplicitata nei diversi modi • Compiere semplici considerazioni sull'utilizzo di circuiti integrati • Individuare le alee statiche 	LF LT AL	VO VS	12	ST EX
LAB		<p>Verifica sperimentale (simulata) delle porte logiche: Porte AND, OR, NAND, NOR. Progetto e verifica del funzionamento di un circuito logico data la tabella di verità. Progetto e realizzazione (simulata) di un circuito logico data la tabella di verità ecc.). La principale componentistica integrata disponibile in scala MSI Codificatori, decodificatori, multiplexer (MPX), demultiplexer (DPX),</p>	<p>Saper:</p> <ul style="list-style-type: none"> • Sviluppare in tutte le sue fasi l'esperienza proposta secondo le specifiche assegnate (es. analisi, ricerca e lettura di documentazione tecnica in lingua italiana e inglese, calcoli preliminari, progetto, simulazione, collaudo, correzione, taratura, misure sul circuito, raccolta dei dati dell'esperienza, ecc) • Riconoscere ed utilizzare la simbologia prevista. • Utilizzare e produrre documentazione tecnica. 	PC AL	VP	10	ST EX PT

Argomento	2	Circuiti con memoria (sequenziali)					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 Definizioni e concetti fondamentali. Componenti con memoria	Elettronica di base in DC Argomenti ed unità precedenti e relativi prerequisiti	Circuiti sequenziali: definizioni e concetti fondamentali, il grafo, diagramma temporale, la tabella delle transizioni. Confronto con i circuiti combinatori. Latch e Flip-Flop (FF): generalità, il latch SR, latch SR con enable, il latch D, la logica temporizzata, FF-D, FF-JK, FF-T, positive/negative edge-triggered e Master-Slave, trasformazioni tra flip-flop. Il grafo e le tabelle di funzionamento, i tipici segnali di controllo, ingressi diretti asincroni, caratteristiche temporali dei FF e semplici	Saper: <ul style="list-style-type: none"> Leggere, interpretare e comprendere il grafo, la tab. delle transizioni ed il diagramma temporale di un FF e di un dato circuito sequenziale e viceversa saperli tracciare in base ad un dato FF e/o circuito sequenziale. Comprendere, analizzare descrivere il funzionamento e progettare semplici circuiti sequenziali (es. generici automi, contatori a modulo qualunque, divisori di frequenza, registri, ecc) correttamente inizializzati ed interfacciati. 	LF AL LG LM EC	VS	10	ST EX
U2 Circuiti sequenziali. Concetti di circuiti asincroni e sincroni.	Argomenti ed unità precedenti e relativi prerequisiti	Concetti di circuiti sincroni e asincrono. Contatori, circuiti di conteggio, memorizzazione e visualizzazione. Contatori sincroni e asincroni modulo 2^n e modulo qualunque, contatori Up, Down e Up/Down, contatori ad anello. Realizzazione con i FF. Registri a scorrimento e ad anello, SISO, SIPO, PISO, PIPO, realizzazione con i FF.	Saper: Realizzare i circuiti sequenziali studiati con porte logiche elementari. Utilizzare i dispositivi sequenziali integrati studiati per la realizzazione di semplici circuiti applicativi. Comprendere, interpretare, analizzare e valutare schemi che comprendono circuiti sequenziali. Comprendere, analizzare descrivere e progettare: <ul style="list-style-type: none"> il funzionamento di contatori a modulo qualunque o divisori di frequenza, correttamente inizializzati ed interfacciati. i principali tipi di registri 	LT MA DI AL	VS	10	ST EX

U3 Introduzione automi agli stati finiti		Definizione schematica e formale della struttura di un automa; stati, ingressi, uscite. Automa di Moore e automa di Mealy. Rappresentazione grafica degli automi di Moore e di Mealy. Grafo degli stati, rappresentazione tabellare ed implementazione binaria. Determinazione Funzioni Transizioni di Stato (FTS) e Funzioni Trasformazione Uscite (FTU) Implementazione di semplici automi.	Saper: Interpretare la rappresentazione grafica dell'Automa di Moore partendo da casi pratici concreti Determinare, da grafo degli stati, usando la metodologia della macchina "Automa di Moore/Mealy", la relativa tabella degli stati e viceversa. Eseguire la relativa implementazione binaria Individuare, in forma canonica, le FTS e le FTU	LT MA DI AL	VS	5	ST EX
LAB		Semplici applicazioni dei circuiti di conteggio e/o memoria. Progetto e simulazione di un automa. applicazioni. Circuito di inizializzazione RC e antirimbando.	Saper: Sviluppare in tutte le sue fasi l'esperienza proposta secondo le specifiche assegnate. Comprendere e consultare i manuali e datasheets ed estrarre i parametri principali di un circuito sequenziale.	LA PC AO	VP	10	ST EX PT

Argomento 3		Elementi di informatica					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 I numeri ed i codici	Concetti elementari di elettronica Struttura tipica di un sistema di elaborazione Matematica e fisica del biennio Tecnologie informatiche del biennio	Sistema di numerazione decimale, binario, ottale, esadecimale ed in base qualsiasi. Concetto di bit, byte e word. Conversioni tra le diverse basi ed operazioni aritmetiche. Rappresentazione e conversione dei decimali. Overflow e Underflow nelle operazioni con e senza segno. Numeri relativi: modulo e segno e complemento a due. Codici: ASCII (e concetto di bit di parità), Gray e BCD. Proprietà ed impieghi dei diversi i codici. Esercizi	Saper: Operare la conversione tra le diverse basi di numerazione di numeri interi e con decimali e compiere operazioni aritmetiche. Convertire un numero con segno in un numero binario in complemento a due o modulo-segno e viceversa. Compiere operazioni di somma o sottrazione con i numeri con segno. Riconoscere le situazioni di overflow e underflow nelle operazioni a lunghezza fissa con o senza segno. Convertire un numero/simbolo come previsto dalle specifiche dei diversi codici e viceversa.	LF LI EC LT DI AL	VS	10	ST EX

U2 Il computer ed il software		<p>Configurazione fondamentale di un PC: lo schema a blocchi ed il suo funzionamento, la struttura hardware di un P.C. e le funzioni delle singole parti e le relazioni reciproche.</p> <p>Livelli di descrizione del funzionamento di un sistema e di funzionalità di un computer.</p> <p>Livelli dei linguaggi di programmazione: strutturazione del software di un P.C. e le funzioni delle singole parti e le relazioni reciproche.</p> <p>Interprete e compilatore.</p>	<p>Saper:</p> <p>Riprodurre la struttura per schema a blocchi di un elaboratore (aspetti hardware e software) e saper descrivere la funzione svolta nell'insieme, da ciascuna parte e le reciproche relazioni intercorrenti.</p> <p>Interpretare correttamente la sequenza delle fasi di accensione e spegnimento di un P.C., avvio e chiusura di una applicazione.</p> <p>Riconoscere vantaggi e svantaggi nell'uso di un compilatore o interprete ed i migliori campi di impiego.</p>	LF LI EC LT DI AL	VP	5	ST EX
U3 Algoritmi e analisi dei problemi.	Argomenti ed unità precedenti e relativi prerequisiti	<p>Il concetto di algoritmo e di programma.</p> <p>La tecnica di rappresentazione mediante flowchart e pseudocodice.</p> <p>Le strutture di controllo.</p> <p>Gli algoritmi strutturati e non. Proprietà.</p> <p>Analisi dei problemi: progettazione TOP-DOWN e BOTTOM-UP. Semplici criteri di progetto.</p> <p>Semplici criteri di progetto dell'algoritmo.</p> <p>Esempi ed esercizi su problemi reali.</p>	<p>Saper:</p> <p>Analizzare nei minimi dettagli un problema anche complesso e tradurlo in una procedura codificata.</p> <p>Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in pseudocodice e/o mediante un flow-chart</p> <p>Riconoscere, interpretare ed utilizzare il linguaggio e la simbologia impiegata per descrivere un algoritmo.</p> <p>Comprendere un assegnato l'algoritmo rappresentato mediante le forme studiate.</p> <p>Riconoscere se un algoritmo è dato in forma strutturata o meno e saperlo strutturare.</p> <p>Analizzare e valutare la correttezza e l'ottimizzazione di un algoritmo in base ai criteri studiati.</p>	LF LI EC LT DI AL	VS VP	10	ST EX PR PG

Argomento	4	Programmazione in C/C++					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 Introduzione alla programmazione in C/C++	Argomenti ed Unità precedenti e relativi prerequisiti.	<p>Editor, compilatore, interprete, linker, debugger. I codice sorgente, oggetto ed eseguibile, le librerie. Help.</p> <p>Le regole per la stesura del software. I commenti.</p> <p>Errori logici e sintattici.</p> <p>La funzione main, la comunicazione con l'utente (printf/scanf e/o cin/cout, l'insieme delle funzioni get e put).</p> <p>Specificazione del formato.</p> <p>Variabili e costanti, gli identificatori, le parole chiave.</p> <p>Le istruzioni per le strutture sequenziali, iterative, condizionali, di selezione semplice e multipla. Strutture nidificate. L'istruzione nulla e composta. Istruzioni di salto.</p>	<p>Saper:</p> <p>Scegliere i tipi di dati più opportuni per risolvere un determinato problema.</p> <p>Scrivere, compilare, aggiungere le librerie e produrre un file eseguibile di un programma in C/C++.</p> <p>Effettuare il debug di un programma.</p> <p>Effettuare una ricerca mediante l'help dell'ambiente ed i manuali.</p> <p>Scrivere un algoritmo dato in pseudocodice o flowchart nel corrispondente programma in linguaggio C/C++ e viceversa.</p> <p>Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in linguaggio C/C++.</p> <p>Utilizzare le principali funzioni e strumenti dell'ambiente di sviluppo, saperne impostare le opzioni e le proprietà.</p>	<p>LF</p> <p>LI</p> <p>LG</p> <p>LM</p> <p>EC</p>	<p>VS</p> <p>VP</p>	<p>10</p>	<p>ST</p> <p>EX</p> <p>PR</p> <p>PG</p>
U2 Procedure e funzioni	Argomenti ed Unità precedenti e relativi prerequisiti.	<p>Introduzione. Dichiarazione e definizione, chiamata ed esecuzione di una funzione.</p> <p>Vantaggi e svantaggi nell'uso delle funzioni.</p> <p>Variabili globali e locali.</p> <p>Parametri e valore di ritorno di una funzione.</p> <p>I prototipi, il preprocessore e le direttive di assemblaggio.</p> <p>Struttura di un programma.</p> <p>Dichiarazione di variabili e funzioni nei diversi livelli. Inizializzazione e classi di memorizzazione</p>	<p>Saper:</p> <p>Scegliere correttamente gli elementi studiati (tipi di funzione, tipi di variabili, direttive, classe di memorizzazione ecc.) più opportuni per risolvere un determinato problema.</p> <p>Creare una libreria di funzioni utente atta alla soluzione di applicazioni di casi reali.</p>	<p>LT</p> <p>MA</p> <p>DI</p> <p>AL</p>	<p>VS</p> <p>VP</p>	<p>10</p>	<p>ST</p> <p>EX</p> <p>PR</p> <p>PG</p>

U3 Tipi aggregati		<p>Vettori (array) e matrici bi-multi dimensionali. Operazioni sui vettori.</p> <p>Vettori di caratteri e stringhe, vettori e funzioni.</p> <p>Dichiarazione di tipi.</p> <p>Strutture, unioni, enumerazioni.</p> <p>La rappresentazione interna dei vari tipi, la dichiarazione, l'inizializzazione, la lettura e la scrittura, gli impieghi e le relative parole chiave del linguaggio.</p>	<p>Saper:</p> <p>Rappresentare, definire e dichiarare i tipi di dati studiati.</p> <p>Scegliere correttamente gli elementi studiati (tipo di dato strutturato, operatori, parole chiave ecc.) più opportuni per risolvere un determinato problema.</p>	LA VP LL CD	VS VP	5	ST EX PR PG
U4 File e stream (Gestione stringhe e File)		<p>Introduzione. Files di testo e binari. Files e flussi.</p> <p>Accesso, apertura, scrittura, lettura, chiusura di un file.</p> <p>Stream di input/output, input e output non formattati, accessi sequenziali e random, stream come argomenti di funzioni.</p> <p>Svuotamento di uno stream ed eliminazione di un file.</p>	<p>Saper:</p> <p>Aprire, scrivere, leggere, chiudere files e stream ed accedere nelle diverse modalità.</p> <p>Scegliere correttamente gli elementi studiati (tipo di elemento, funzioni, operatori, modalità, tipo di accesso ecc.) più opportuni per risolvere un determinato problema.</p> <p>Saper scrivere e leggere un file di testo e/o binario.</p>	LA VP LL CD	VS VP	5	ST EX PR PG
U5 Reference e puntatori		<p>Introduzione. Reference e puntatori. Impiego nel passaggio dei parametri e nell'elemento di ritorno delle funzioni. Passaggio per valore.</p> <p>Aritmetica dei puntatori.</p> <p>Puntatori a dati di tipo aggregato.</p> <p>Parametri e valori di ritorno del main.</p>	<p>Saper:</p> <p>Rappresentare, definire e dichiarare reference e puntatori.</p> <p>Scegliere correttamente gli elementi studiati (variabile, reference, puntatore, operatori, parole chiave ecc.) più opportuni per risolvere un determinato problema.</p>	LA VP LL CD	VS VO	3	ST EX PR PG
Laboratorio	Argomenti ed Unità precedenti e relativi prerequisiti	<p>Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati.</p> <p>Progettazione codice per l'automazione di problemi di E&E;</p> <p>Automati contatori in logica programmata;</p> <p>Progettazione di librerie con funzioni utente</p>	<p>Saper:</p> <ul style="list-style-type: none"> • progettare l'algoritmo per la soluzione di problemi reali • realizzare il codice C/C++ di un algoritmo; • Realizzare il codice di funzioni utente da richiamare a bisogno; • Effettuare il collaudo del software • Realizzare documentazione di rito. 	LT VP LL PC	VP	12	

Argomento	5	Programmazione con Arduino					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 Struttura hardware e IDE di Arduino	Argomenti ed Unità precedenti e relativi prerequisiti.	Architettura e tipi di schede Arduino. La IDE di programmazione della scheda: Input/Output	Saper: descrivere l'architettura di una scheda Arduino Saper descrivere ed usare l'ambiente di sviluppo della scheda su Windows e/o altri S.O.	LF LI LG LM EC	VS VP	5	ST EX PR PG
U2 Programmazione della scheda Arduino		Generalità sulla programmazione in C di Arduino. Tipi di istruzioni, struttura di un programma, funzione dei pin, cicli, cicli nidificati, subroutine, strutture condizionali, confronti, timer.	Saper padroneggiare le analogie tra la programmazione in C e l'IDE di Arduino Saper elaborare del software applicativo, su specifica, per semplici applicazioni digitali.	LT MA DI AL	VS VP	15	ST EX PR PG
Laboratorio		Simulazione scheda Arduino con Proteus/Tinkercad Piccole automazioni digitali di pilotaggio di led Verifica sperimentale, tramite software e breadboard delle porte logiche: porte AND, OR, NAND, NOR. Progetto e verifica del funzionamento di un circuito logico programmato, data la tabella di verità. Visualizzazione di un conteggio da 0 a 9 su un display a LED 7 segmenti, mediante la programmazione della scheda.	Saper: Realizzare semplici automatismi digitali (giochi di luci) con Arduino e scheda breadboard Saper sviluppare la logica dei circuiti combinatori nella logica programmata di Arduino. Utilizzare simulatori quali Proteus, Tinkercad, Multisim	LA PC AO	VP	10	ST EX PT

Argomento	6	Programmazione con LabView					
Unità	Prerequisiti	Conoscenze	Abilità	Metodi e mezzi	Verifiche	Tempi	Lavoro domestico
U1 Introduzione alla programmazione con LabView	Argomenti ed Unità precedenti e relativi prerequisiti.	Introduzione all'ambiente LabView: il VI ed i sub-VI, controlli e indicatori, le palette, gli strumenti, i concetti di connettore ed icona. Aprire, chiudere, salvare (eventualmente con opzioni ed in librerie) un VI. Gestione degli aspetti grafici (colore, dimensioni, forma, nome, scala ecc. degli oggetti, documentazione e tips), etichette ed elementi grafici (linee, rettangoli, ecc).	Saper: Riconoscere controlli ed indicatori, tipo, caratteristiche e proprietà e saperli utilizzare ed impostare. Scrivere, valutare ed interpretare correttamente le espressioni contenenti i diversi tipi di dati, funzioni/VI ed operatori studiati. Scrivere un algoritmo dato in pseudocodice o	LF LI LG PC	VS VP	10	ST EX

		<p>Proprietà dei VI (aspetto, documentazione, revisioni ecc). Gerarchia dei VI.</p> <p>Debug del software: esecuzione passopasso con visualizzazione del flusso dati, probe, breakpoints.</p> <p>Help: Manuali utente, help, context help, forum e materiali reperibili in rete.</p> <p>Tipi di dati (semplici ed aggregati) associabili a variabili, controlli ed indicatori.</p> <p>VI ed operatori sui diversi tipi di dati aggregati e non.</p>	<p>flowchart nel corrispondente programma in LabView e viceversa.</p> <p>Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in LabView.</p> <p>Scegliere correttamente gli elementi studiati più opportuni per risolvere un determinato problema.</p> <p>Leggere, interpretare, comprendere in tutte le sue parti, proprietà ed opzioni un VI.</p> <p>Tracciare in modo completo, mandare in esecuzione nelle diverse modalità, gestire le librerie di un programma in LabView per mezzo dell'elaboratore.</p> <p>Effettuare il debug di un programma.</p> <p>Effettuare una ricerca mediante l'help dell'ambiente ed i manuali.</p> <p>Utilizzare le principali funzioni e strumenti dell'ambiente di sviluppo, saperne impostare le opzioni e le proprietà.</p>				
U3 <i>Le strutture ed i cicli</i>		<p>Le strutture fondamentali (for, while, case, sequenze ecc.). Tunnel, indexing e shift register.</p> <p>Formula node ed expression node.</p> <p>La gestione del tempo ed i ritardi temporali.</p>	<p>Saper effettuare una ricerca mediante l'help dell'ambiente ed i manuali.</p> <p>Saper utilizzare le principali funzioni e strumenti dell'ambiente di sviluppo e saperne impostare le opzioni e le proprietà.</p>	LT DI AL PC	VS VP	10	ST EX
Laboratorio	Argomenti ed Unità precedenti e relativi prerequisiti.	<p>Impostazione dell'ambiente di sviluppo.</p> <p>Stesura, esecuzione e debug di programmi in cui sono presenti gli elementi studiati.</p> <p>Piccole automazioni digitali di pilotaggio di led</p> <p>Verifica sperimentale, tramite software, VI, delle porte logiche: porte AND, OR, NAND, NOR.</p> <p>Progetto e verifica del funzionamento di un circuito logico programmato, data la tabella di verità.</p> <p>Visualizzazione di un conteggio da 0 a 9 su un display a LED 7 segmenti, mediante la programmazione di un VI.</p>	<p>Saper realizzare semplici automatismi digitali con LabView</p> <p>Saper sviluppare la logica dei circuiti combinatori nella logica programmata di LabView</p>	LA PC AO PC	VS VP	10	ST EX PT

Esperienze pratiche previsto per ogni modulo:

Modulo	Argomento	Esperienza pratica
1	Circuiti combinatori	Uso algebra booleana e mappe di Karnaught per implementare controllo display a 7 segmenti 0-9 dec, 0-F hex
2	Circuiti con memoria (sequenziali)	Semplici automi contatori di MOORE in logica cablata e logica programmata
3	Elementi di informatica	Gestione ed implementazione di algoritmi
4	Programmazione in C/C++	Gestione di Array: Codice per calcolare media tra più materia di una classe e ricerca del voto massimo per materie e in assoluto Progetto e verifica del funzionamento di un circuito logico programmato, data la tabella di verità. Visualizzazione di un conteggio da 0 a 9 su un display a LED 7 segmenti, mediante la programmazione di un VI
5	Programmazione con Arduino	Progetto e verifica del funzionamento di un circuito logico programmato, data la tabella di verità. Visualizzazione di un conteggio da 0 a 9 su un display a LED 7 segmenti, mediante la programmazione di un VI
6	Programmazione con LabView	Progetto e verifica del funzionamento di un circuito logico programmato, data la tabella di verità. Visualizzazione di un conteggio da 0 a 9 su un display a LED 7 segmenti, mediante la programmazione di un VI

NOTE

LEGENDA

Colonna 2° Unità: Ux = Unità didattica numero x

Colonna 5° **Metodi e mezzi:**

Metodo:

LF = Lezione frontale
LI = Lezione interattiva
LG = Lavoro di gruppo
LM = Lavoro manuale o pratico
EC = Esercizi in classe

Supporto didattico: LT = Libro di testo

MA = Manuali tecnici del
laboratorio.

DI = Dispense o materiali scaricabili
dalla rete (es. datasheets)

AL = Appunti della lezione

Supporto tecnico: LA = Laboratorio

VP = Videoproiettore

LL = Lavagna luminosa

CD = Materiali in CD-Rom

PC = Personal Computer, relativi
pacchetti applicativi, internet.

AO = Attrezzatura ordinaria del
laboratorio

Colonna 6° Verifiche

S = Scritta

O = Orale

Colonna 7° **Tempi (ore):**

PQ = Primo quadrimestre

SQ = Secondo quadrimestre

Colonna 8° **Lavoro domestico:**

ST = Studio teorico

EX = Esercizi

PT = Produzione tesine, relazioni

Colonna 9° **Competenze in uscita interessate:**

i numeri fanno riferimento alle
competenze in uscita riportate a
pag. 5
