



ISTITUTO "MAX PLANCK"
ISTITUTO TECNICO E LICEO SCIENTIFICO DELLE S.A.
VIA FRANCHINI, 1 31020 - LANCENIGO DI VILLORBA (TV) C.M. TVTF04000T - C.F. 94000960263 - TEL. 0422 6171 R.A.



PROGRAMMAZIONE DI DIPARTIMENTO

Sistemi Automatici
per Elettronica
3^a

Classe: 3^a
Indirizzo: Elettrotecnica ed Elettronica
Articolazione: Elettronica
Materia: Sistemi automatici
A.S.: 2024-25

Finalità Educative.

Come recita il P.T.O.F.:

“Gli allievi non saranno solo oggetto dell’azione insegnamento/apprendimento, ma “co-protagonisti” di essa, pertanto, saranno adeguatamente guidati a:

Mettersi alla prova, ad autovalutarsi con senso critico e consapevolezza di sé con conseguente arricchimento della personalità;

Acquisire competenze metodologiche e formative per un consapevole inserimento civile e sociale;

Acquisire un adeguato senso di responsabilità (diritti/ doveri, lealtà, impegno, frequenza);

Costruire una base valoriale comune in cui riconoscersi (abitudine al vivere civile, attenzione alle problematiche del mondo, consapevolezza che la diversità è fonte di arricchimento e rifiuto dei pregiudizi).

“L’istituto, nel proprio disegno educativo, promuove la formazione completa della persona, nel rispetto dei principi fondamentali sanciti dalla Costituzione della Repubblica italiana, con riferimento soprattutto agli artt. 2, 3, 9, 11, 33, 34, e ai diritti umani, sanciti dal diritto internazionale”.

Il Dipartimento di Elettronica ed elettrotecnica fa proprie le indicazioni del P.T.O.F. cercando di sviluppare e potenziare negli allievi i seguenti valori:

- La responsabilità (verso di sé e verso gli altri) collegata necessariamente alla libertà, che la rende possibile, e al rispetto dei diritti altrui.
- L'impegno profuso nel miglioramento di sé e degli altri che dà luogo al merito.
- La solidarietà nei confronti dei soggetti più deboli, pur senza “protezionismi”.
- La giustizia, che, fondandosi sul riconoscimento dell'uguaglianza dei diritti e dei doveri di tutti, nel rispetto delle regole, rappresenta al tempo stesso anche un esercizio di cittadinanza.
- La pace, intesa come rispetto delle posizioni di tutti e ripudio dell'intolleranza, anche in una prospettiva di dialogo interculturale e di multiculturalità.
- L'ambiente, nella consapevolezza che il pianeta Terra è patrimonio universale dell'umanità e delle generazioni future.

Competenze cognitive.

Il Dipartimento di Elettronica ed Elettrotecnica cercherà, inoltre, 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 tre punti esplicitati nella tabella seguente. Per ciascuno dei tre punti sono stati messi in evidenza i livelli in uscita per le due classi del secondo biennio e per l'ultimo anno.

Imparare ad imparare

- *“Organizza il proprio apprendimento valutando: tempi, strategie, modalità”*
 - Classe 3^a - Rispetta i tempi di consegna sia dei lavori in classe sia dei lavori domestici.
 - Classe 4^a - Rispetta tempi di consegna e modi di esecuzione dei lavori.
 - Classe 5^a - Sa gestire in modo autonomo il proprio lavoro in classe e domestico dominando le strategie più opportune per portare a termine i compiti assegnati nella maniera migliore.
- *“Utilizza la lingua scritta per attività di studio (appunti, riassunti, schemi, schedature, mappe”*
 - Classe 3^a - Sa prendere appunti in maniera precisa e puntuale. Sa commentare adeguatamente le soluzioni proposte negli elaborati scritti.
 - Classe 4^a - Sa sintetizzare nelle varie forme utilizzando anche schemi, diagrammi, grafici.
 - Classe 5^a - Sa ottimizzare e integrare l'uso di appunti e schemi anche con l'uso di sistemi informatici.

Comunicare efficacemente

- *“Pianifica ed organizza l'esposizione orale tenendo conto del destinatario, della situazione comunicativa, delle finalità, dei tempi.”*
 - Classe 3^a - Sa pianificare ed organizzare l'esposizione orale tenendo conto del destinatario.
 - Classe 4^a - Sa pianificare e organizzare l'esposizione orale tenendo conto delle finalità.
 - Classe 5^a - Sa pianificare e gestire in modo autonomo l'esposizione orale tenendo conto della destinazione, delle finalità e dei tempi a disposizione.

Finalità ed obiettivi generali

Come si evince dalle linee guida ministeriali, il corso di Sistemi automatici è finalizzato a far conseguire allo studente, al termine del percorso quinquennale, 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 attività lavorativa;
- saper interpretare il proprio autonomo ruolo nel lavoro di gruppo;
- essere consapevole del valore 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 ed obiettivi specifici

La disciplina, nell'ambito della programmazione del Consiglio di Classe (*CdC*), concorre in particolare al raggiungimento dei seguenti risultati di apprendimento, relativi all'indirizzo, espressi in termini di competenza:

- utilizzare la strumentazione di laboratorio e di settore e applicare i metodi di misura per effettuare verifiche, controlli e collaudi
- utilizzare linguaggi di programmazione, di diversi livelli, riferiti ad ambiti specifici di applicazione
- analizzare il funzionamento, progettare e implementare sistemi automatici
- analizzare il valore, i limiti e i rischi delle varie soluzioni tecniche per la vita sociale e culturale con particolare attenzione alla sicurezza nei luoghi di vita e di lavoro, alla tutela della persona, dell'ambiente e del territorio.
- redigere relazioni tecniche e documentare le attività individuali e di gruppo relative a situazioni professionali

Programmazione.

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

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

Verifiche e valutazioni

La disciplina prevede voto unico come deliberato dal Collegio dei Docenti (*CdD*).

Il numero minimo di verifiche è di due per il primo quadrimestre e tre nel secondo quadrimestre, scelte tra le tipologie previste.

Le tipologie di verifica sono tutte quelle previste dal P.T.O.F., in particolare:

- Verifica orale: individuali o di gruppo, ad es. con domande a risposta chiusa o aperta con risoluzione di problemi semplici o articolati, con calcoli di progetto, analisi e sintesi e valutazione di algoritmi e codice, realizzazione di circuiti, assemblaggio di dispositivi, tracciatura di schemi, grafici e diagrammi, lettura e comprensione di fogli tecnici ecc. Le prove potranno essere assolve anche in forma scritta o al computer;
- Verifica scritta: tutte le tipologie previste dal P.T.O.F. quali ad es. prove strutturate, semi-strutturate, questionari, domande a risposta chiusa o aperta con risoluzione di problemi semplici o articolati, con calcoli di progetto, tracciatura di grafici e diagrammi, analisi e sintesi e valutazione di algoritmi e codice, lettura e comprensione di fogli tecnici ecc. Prova al computer;

- Verifica pratica / grafico-pratica / relazioni: tutte le tipologie previste dal P.T.O.F., individuali o di gruppo, ad es. realizzazione del lavoro, collaudo hardware e/o software, ricerca e correzione errori e guasti, cablaggio del circuito, misure sul circuito, riparazione ecc. in funzione della natura del progetto, documentazione che accompagna il progetto e/o la realizzazione pratica;
- Il controllo dei quaderni, il lavoro assegnato per casa ed altri elaborati;

Ciascuna prova di laboratorio è 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.

Le valutazioni delle parti grafiche e pratiche possono essere effettuate, oltre che al termine del lavoro, anche in fasi intermedie ed eventualmente integrate da verifiche orali.

Le prove di laboratorio consistono nella realizzazione dell'esperienza assegnata (circuito e/o sistema e/o software ecc.), nel collaudo e relativa correzione, ed eventualmente nella produzione della relazione cartacea sulla conduzione e risultati della prova pratica in laboratorio.

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

Considerata la compresenza di due insegnanti le ore del laboratorio sono utilizzate inoltre come sportello per il ripasso curricolare per gli studenti che lo richiedono compatibilmente con le altre esigenze didattiche e le specifiche indicazioni del C.d.C.

Le parti facoltative di approfondimento per le eccellenze vertono generalmente su argomenti trattati durante l'anno scolastico e sono di solito concordate con gli studenti per tener conto delle loro inclinazioni ed interessi. La valutazione è effettuata sulla base di una interrogazione orale e/o una produzione pratica e/o un elaborato.

Per quanto concerne la valutazione, oltre a rimandare a quanto stabilito dal Collegio dei Docenti, dal P.T.O.F. e dal Consiglio di Classe in apposite riunioni e redatto nei relativi verbali, si fa riferimento alla tabella sotto riportata (*Descrittori dei voti delle prove di verifica*).

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).

Le prove di verifica grafiche/pratiche non consegnate in alcuna loro parte nei tempi previsti sono valutate, in assenza di comprovate e valide motivazioni che abbiano impedito lo svolgimento, con il minimo dei voti della tabella di valutazione del profitto. Ritardi di lieve entità comportano una valutazione che non può essere né ottima né eccellente.

La valutazione della parte pratica, nel caso in cui l'allievo operi o si appresti ad operare in condizioni che non sono di sicurezza, non può essere in alcun modo sufficiente e varia in base alla gravità della violazione dei regolamenti.

In presenza di risultati non soddisfacenti nelle prove pratico-grafiche che si sviluppino nel lungo periodo con parti che possono essere svolte o completate a casa, in caso di recupero e qualora le condizioni didattiche lo permettano, possono essere recuperate una sola volta sottoponendosi alla relativa prova con un nuovo problema assegnato (rifacimento completo del lavoro con valutazione sulla intera scala dei voti) oppure sullo stesso problema (con voto massimo 7 in caso di prestazione molto positiva).

Dopo ogni verifica, in particolar modo quella orale, lo studente è invitato ad autovalutarsi affinché si abitui all'analisi delle proprie prestazioni, a valutare i propri punti di forza e di debolezza in modo da poter raggiungere consapevolmente i propri obiettivi.

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 Dipartimento e dal P.T.O.F.				
Voto	Giudizio	Conoscenza	Abilità	Competenze
10	Eccellente	Completa, precisa ed approfondita	<u>Esposizione organica e originale.</u> <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	Ottimo	Completa e precisa	<u>Esposizione organica, completa e precisa.</u> <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	Buono	Completa	<u>Esposizione completa.</u> <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	Discreto	Abbastanza Completa e sostanzialmente sicura	<u>Esposizione chiara e ordinata.</u> <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	Sufficiente	Essenziale degli elementi principali della disciplina	<u>Esposizione chiara e semplice.</u> <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	Insufficiente	Superficiale, in presenza di errori	<u>Esposizione generica e stentata.</u> <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	Grave insufficienza	Frammentaria con errori rilevanti	<u>Esposizione incerta e disorganica.</u> <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	Insufficienza molto grave	Frammentaria e lacunosa degli elementi con errori gravi e diffusi	<u>Esposizione stentata, confusa e disorganica.</u> <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 nullo	Quasi completamente errata	<u>Esposizione confusa.</u> <u>Linguaggio</u> approssimativo. <u>Comprensione e risoluzione</u> logico-operative assenti di procedure tecniche disciplinari.	Quasi assente
1	Nullo	Completamente errata	<u>Esposizione decisamente confusa.</u> <u>Linguaggio</u> decisamente approssimativo. <u>Comprensione e risoluzione</u> logico-operative assenti di procedure tecniche disciplinari.	Assente

* Qualora si verificasse asimmetria tra i livelli dei descrittori, si adotta il criterio della prevalenza, a condizione che siano acquisite le conoscenze minime.

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.

Si ritiene che un allievo che segua con la dovuta serietà ed attenzione il lavoro svolto in classe nell'orario curricolare, per poter conseguire una preparazione ed un profitto sufficienti abbia la necessità di impegnarsi settimanalmente nello studio e nel lavoro domestico per almeno 3h.

Per ottimizzare l'impiego del tempo le verifiche orali si possono svolgere durante le attività di laboratorio.

Le ore di lezione extracurricolare (sportello pomeridiano o "SOS") sono indicate in appositi registri predisposti dalla scuola.

Testo adottato

Cerri, Ortolani, Venturi– "Nuovo Corso di Sistemi Automatici 1" – Hoepli

Sono da considerare parte integrante i materiali in forma digitale previsti dal testo in adozione forniti dall'editore e che gli allievi devono recuperare autonomamente.

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 – Art. Elettronica – Ed. Tramontana

Sono da considerare parte integrante i materiali in forma digitale previsti dal testo in adozione forniti dall'editore e che gli allievi devono recuperare autonomamente.

Altri strumenti e sussidi didattici

Gli appunti di lezione, i manuali digitali e cartacei del laboratorio, le dispense ad uso interno e gli esercizi e questionari per il ripasso dei contenuti della materia e per la preparazione alle prove, i fogli tecnici, i manuali, le simulazioni e quant'altro reso disponibile dall'insegnante in classe o attraverso la piattaforma informatica di istituto (Rete Interna, Internet, Moodle, PlanckSuite, strumenti informatici per la didattica resi disponibili da maxplanck.edu.it, ecc).

Alcuni docenti utilizzano una classe virtuale, raggiungibile dal sito della scuola, di cui tutti gli allievi dispongono delle credenziali e a cui devono fare riferimento per le attività didattiche (reperimento materiali, eventuali comunicazioni docente-studente, inserimento materiali, test on-line, ecc.) con frequenza non meno di due volte alla settimana e soprattutto quando l'insegnante lo richiede esplicitamente.

I software indicati nella programmazione.

I fogli tecnici dei componenti e/o degli strumenti e software utilizzati nel laboratorio, generalmente reperibili dalla rete e consultabili in formato elettronico.

Criteri di verifica e feedback

Verifica annuale del realizzato in relazione al progettato, con l'analisi dei relativi risultati in termini di profitto, capacità e competenze acquisite. In base ai risultati della verifica si procederà a eventuali modificazioni o integrazioni del presente piano di lavoro.

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 sono svolti a lezione pur rispettandone le necessarie propedeuticità.
- La durata effettiva delle spiegazioni e delle esercitazioni dipendono 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, svolte nelle ore di compresenza, non è incorporato. Sarà indicato solo nel caso in cui il ripasso riguardi l'intera classe.
- Verifiche (es. orali) ed attività di recupero-ripasso sono da intendersi come un'attività strettamente legate 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 intera. Inoltre durante il ripasso possono svolgersi domande di tipo formativo.

- 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, è utilizzata per approfondire e consolidare, attraverso attività pratiche, esercitazioni, simulazioni ecc., le parti trattate in classe. Pertanto tali attività seguono di norma, nei contenuti, nelle esperienze e nelle eventuali verifiche, l'attività teorica ed il tipo ed il numero di esperienze di laboratorio non è specificato perché variabile dipendente dalle contingenze didattiche.
- Oltre al testo in adozione gli allievi sono tenuti ad utilizzare e a conservare in modo ordinato le dispense, quesiti, problemi ecc. forniti dai docenti, oltre alla cura degli appunti di lezione in modo da agevolare lo studio domestico.
- L'attività di laboratorio è strettamente connessa ai contenuti teorici dell'argomento corrispondente. Le conoscenze e abilità/competenze delle parti teoriche sono valutate anche nelle attività di laboratorio corrispondenti.
- In base a quanto deciso dal C.d.D. e/o C.d.C., e nei termini previsti, le lezioni possono essere trasmesse o integrate mediante le modalità sincrona, con interazione in tempo reale docenti/discenti, o asincrona, ad es. mediante studio di materiali forniti dagli insegnanti o presenti in rete, lavori o esercitazioni, risoluzione di problemi a breve o lungo termine individuali o di gruppo, visione di video, analisi, sintesi o valutazione di progetti e relativa produzione dei materiali richiesti dai docenti, ecc.
- La presente programmazione può subire modifiche in base alle necessità emergenti e non previste o prevedibili.
- Nella programmazione di ogni Modulo/Unità non si riportano i contenuti degli argomenti precedenti già svolti e che possono essere inclusi naturalmente nella trattazione, nei quesiti, problemi, esercitazioni, verifiche, relativi all'argomento in corso.

Abilità \ Competenze generali comuni a tutte le unità del programma, dei contenuti irrinunciabili e conoscenze e competenze minime per l'accesso alla classe successiva.

Le seguenti competenze generali si intendono riferite a tutti gli argomenti e unità della programmazione, dei contenuti irrinunciabili e delle conoscenze e competenze/abilità minime per l'accesso alla classe successiva 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.
- Effettuare l'analisi, la sintesi e la valutazione degli elementi costituenti le parti trattate (es. sistemi, tecniche, circuiti, programmi ecc.)
- Svolgere semplici calcoli e passaggi matematici, rappresentare mediante testo, grafici, diagrammi, tabelle, schemi, schemi a blocchi, andamenti temporali, relazioni analitiche, ecc. gli elementi studiati, estrarre parametri, proprietà, caratteristiche ecc, saperli interpretare e saper spiegare le relazioni reciproche tra le parti e/o gli elementi trattati.
- Utilizzare correttamente le grandezze studiate e le relative unità di misura.
- 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 C/C++/LabView) senza errori sintattici e logici e, se presenti, saperli individuare e correggere. Viceversa, saper descrivere l'algoritmo implementato da un programma in linguaggio C/C++/LabView, 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 C/C++/LabView e viceversa.
- Scegliere correttamente gli elementi studiati più opportuni per risolvere un determinato problema.
- Scrivere un programma nello specifico linguaggio C/C++/LabView (o altra forma codificata studiata in classe) 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 C/C++/LabView contenente gli elementi studiati per mezzo dell'elaboratore, saperne effettuare il debug e saper utilizzare le principali funzioni dell'ambiente.

- 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.
- Gestire ed impiegare correttamente i materiali e gli strumenti propri e del laboratorio, utilizzare correttamente le postazioni di lavoro.
- Lavorare collaborando attivamente in un gruppo di lavoro. Condividere le informazioni e confrontare il proprio lavoro con il gruppo.
- Utilizzare componenti, dispositivi e strumentazione del laboratorio per realizzare quanto richiesto.
- Scrivere e commentare il codice in modo chiaro, completo e corretto, saperlo collaudare, correggere e discutere.
- Realizzare, collaudare e correggere il circuito ed effettuare le misure richieste.
- Utilizzare pacchetti applicativi per effettuare i calcoli di progetto, tracciare gli schemi elettrici e realizzare la simulazione dei circuiti studiati, scrivere la relazione tecnica.
- Scrivere, compilare, gestire le librerie, effettuare il link e produrre un file eseguibile di un programma in C/C++/LabView 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 C/C++/LabView usando la programmazione TOP-DOWN strutturata.
- Effettuare correttamente una ricerca mediante l'help dell'ambiente, i manuali cartacei, in formato elettronico e online.
- 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/DVD, internet ecc.) e saperli consultare (anche se in lingua inglese).
- Comprendere le indicazioni del produttore per il corretto impiego del componente, software o dispositivo.
- 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).
- Leggere, consultare e comprendere il regolamento di laboratorio, saperlo rispettare, rispettare le indicazioni operative degli insegnanti e del personale della scuola. Saper identificare e segnalare eventuali violazioni. Operare in condizioni di sicurezza.

Programmazione

Unità	U1 - I numeri ed i codici				
Prerequisiti	Concetti elementari di elettrotecnica. Struttura tipica di un sistema di elaborazione. Capacità di comprensione di un testo. Nozioni di base di matematica e fisica e informatica. Utilizzare le principali funzionalità di un P.C., delle periferiche ed degli accessori (es. memorie di massa esterne) e del software di uso comune (videoscrittura)				
Conoscenze	Abilità \ Competenze	Met. e mezzi	Verif	Tempi	Lav. Dom.
<ul style="list-style-type: none"> ▪ Sistemi di numerazione: 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. Codifica in virgola fissa e cenni virgola mobile. Overflow e Underflow nelle operazioni con e senza segno. Bit segnalatori CNZV ▪ Numeri relativi: modulo e segno e complemento a due. Operazioni. ▪ Codici: ASCII esteso e con bit di parità, Gray e BCD. Proprietà ed impieghi dei diversi i codici. ▪ Esempi, esercizi ed applicazioni 	<p>Saper</p> <ul style="list-style-type: none"> ▪ 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 con i numeri con segno. ▪ Riconoscere le situazioni di overflow e underflow nelle operazioni a lunghezza fissa con o senza segno. Impostare i segnalatori CNZV. ▪ Convertire un numero/simbolo come previsto dalle specifiche dei diversi codici e viceversa. 	LF LI LG LM EC LT MA DI AL LA VP TG CD PC AO	VS e/o VO	Tot 17h	ST EX

Prerequisiti		UUPP				
Conoscenze		Abilità \ Competenze	Met. e mezzi	Verifiche	Tempi	Lav. Dom.
Unità	U2 - Il computer ed il software	Saper				
<ul style="list-style-type: none"> ▪ Configurazione fondamentale di un PC: lo schema a blocchi (es. Case, interfacce e periferiche, memoria di massa, CPU, RAM, ROM, I/O, BIOS, bus, clock, ecc) ed il suo funzionamento, la struttura hardware di un P.C. e le funzioni delle singole parti e le relazioni reciproche. Sequenza di boot e shutdown. ▪ Livelli di descrizione di un sistema: del funzionamento e di funzionalità di un computer. ▪ Livelli dei linguaggi di programmazione: dall'alto livello fino al codice macchina. Ambiente di sviluppo. ▪ Software di un P.C. : Strutturazione (S.O., applicativo, drivers) e le funzioni delle singole parti e le relazioni reciproche. Interprete e compilatore, linker, editor, debugger. Funzioni e caratteristiche. ▪ Esempi, esercizi ed applicazioni 		<ul style="list-style-type: none"> ▪ Riprodurre la struttura per schema a blocchi di un elaboratore (aspetti hardware e software) e saperne descrivere le caratteristiche, la funzione svolta nell'insieme, da ciascuna parte e le reciproche relazioni intercorrenti. ▪ Descrivere ed Interpretare correttamente la sequenza delle fasi di accensione e spegnimento di un P.C., avvio e chiusura di una applicazione. ▪ Riconoscere vantaggi e svantaggi nell'uso di un compilatore o interprete ed i migliori campi di impiego. ▪ Saper descrivere le funzioni specifiche di produzione del software ed i vantaggi e svantaggi dei software di diverso livello. 	LF LI LG LM EC			ST EX
Unità	U3 - Algoritmi e analisi dei problemi.	Saper:	LT MA DI AL LA VP TG CD PC AO	VS e/o VO e/o VP	Tot 22 h	ST EX PM
<ul style="list-style-type: none"> ▪ Algoritmo e programma: introduzione e concetti generali. ▪ Le tecniche di rappresentazione: mediante flowchart e pseudocodice. ▪ I blocchi e le strutture di controllo: gli algoritmi strutturati e non. Proprietà. Analisi dei problemi: progettazione TOP-DOWN e BOTTOM-UP. ▪ Criteri di progetto: criteri generali ed ottimizzazioni. . Il tracing di un algoritmo ▪ Esempi, esercizi ed applicazioni <p>Laboratorio</p> <ul style="list-style-type: none"> ▪ Stesura, verifica del funzionamento e correzione di algoritmi realizzati all'elaboratore con opportuni software di tracciatura e simulazione. ▪ (*) 		<ul style="list-style-type: none"> ▪ Analizzare nei minimi dettagli un problema anche complesso e tradurlo in una procedura codificata (flowchart e/o pseudocodice) attraverso le procedure TOP-DOWN e/o BOTTOM-UP strutturate; ▪ Riconoscere, interpretare ed utilizzare i linguaggi e le simbologie impiegate per descrivere un algoritmo. Comprendere un assegnato algoritmo rappresentato mediante le forme studiate e saperlo rappresentare in una qualsiasi altra forma. ▪ Riconoscere se un algoritmo è dato in forma strutturata o meno e saperlo strutturare. ▪ Saper identificare le strutture studiate in un algoritmo, anche complesso, fornito in una qualsiasi forma. ▪ Effettuare l'analisi, la sintesi e la valutazione di un algoritmo. Verificarne la correttezza e l'ottimizzazione in base ai criteri studiati. ▪ Effettuare il tracing di un algoritmo a mano o al simulatore. ▪ Editare, simulare, correggere un algoritmo all'elaboratore. 				

Prerequisiti		UUPP				
Conoscenze		Abilità \ Competenze	Met. e mezzi	Verifiche	Tempi	Lav. Dom.
Unità	U4 - Introduzione alla programmazione in C/C++	Saper: <ul style="list-style-type: none"> Scrivere, valutare ed interpretare correttamente le espressioni contenenti i diversi tipi di dati ed operatori studiati. Scrivere un algoritmo dato in pseudocodice o flowchart nel corrispondente programma in linguaggio C/C++ e viceversa. Scrivere e commentare il codice in modo chiaro, corretto e completo. Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in linguaggio C/C++. Scegliere i tipi di dati più opportuni per risolvere un determinato problema. Rappresentare, definire e dichiarare i tipi di dati studiati. Scrivere, compilare, aggiungere le librerie e produrre un file eseguibile di un programma in C per mezzo dell'ambiente di sviluppo. Effettuare il debug di un programma. Effettuare una ricerca mediante l'help online e dell'ambiente ed i manuali. Utilizzare le principali funzioni e strumenti dell'ambiente di sviluppo, saperne impostare le opzioni e le proprietà. 	LF LI LG LM EC			
	<ul style="list-style-type: none"> Concetti base del C/C++: La funzione main, la comunicazione con l'utente (printf/scanf e/o cin/cout, l'insieme delle funzioni get e put, gets ecc.). Specificazione del formato. Le direttive. I dati: Variabili e costanti, gli identificatori, le parole chiave. Dichiarazione e definizione. L-value e R-value. I tipi fondamentali, le dichiarazioni, le espressioni, gli operatori (aritmetici, logici, bitwise e relazionali, ecc). Precedenze e associatività. Le conversioni. Le regole per la stesura del software. I commenti. Enumerativi. Vita e visibilità, variabili locali e globali, Le strutture di programmazione fondamentali: Le istruzioni per le strutture sequenziali, iterative, condizionali, di selezione semplice e multipla. Strutture nidificate. L'istruzione nulla e composta. Istruzioni di salto. Concetti base di programmazione: Programmi batch e ciclici, le variabili interruttore e di accumulo. Editor, compilatore, linker, debugger. I codice sorgente, oggetto ed eseguibile, le librerie. Impiego dei manuali e dell'Help d'ambiente e on-line. Errori logici e sintattici. Esempi, esercizi ed applicazioni . 		LT MA DI AL	VS e/o VO e/o VP	Unità U4 U5 Tot 32h	ST EX PM
Unità	U5 - Procedure e funzioni	Saper <ul style="list-style-type: none"> Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in linguaggio C/C++. Scegliere correttamente gli elementi studiati (tipi di funzione, tipi di variabili, direttive, classe di memorizzazione ecc.) più opportuni per risolvere un determinato problema. Tracciare un algoritmo nelle diverse forme che impieghi le funzioni. Scrivere, compilare, eseguire, effettuare il debug di un programma che impieghi le funzioni sia nel file principale sia in file esterni. 	LA VP TG CD PC AO			
U4U5 LAB	<ul style="list-style-type: none"> Stesura dell'algoritmo che risolve problemi assegnati contenenti gli elementi studiati. Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati. Compilazione di librerie. Impostazione dell'ambiente e del progetto. (*) 					

Prerequisiti		UUPP				
Conoscenze		Abilità \ Competenze	Met. e mezzi	Verifiche	Tempi	Lav. Dom.
Unità	U6 - Tipi aggregati	Saper: <ul style="list-style-type: none">Rappresentare, definire e dichiarare i tipi di dati studiati.Scegliere correttamente gli elementi studiati (tipo di dato strutturato, operatori, parole chiave ecc.) più opportuni per risolvere un determinato problema.Tracciare algoritmi nelle diverse forme, scrivere, compilare, eseguire, effettuare il debug di programmi, che impieghino le parti studiate.	LF LI LG LM EC LT MA DI AL LA VP TG CD PC AO	VS e/o VO e/o VP	Tot 34h	ST EX PM
<ul style="list-style-type: none">I dati di tipo aggregato: Array e Strutture. Vettori (array) e matrici bi-multi dimensionali. Operazioni con dati aggregati. Vettori di caratteri e stringhe. Strutture. Dichiarazione di tipi.Dati di tipo aggregato contenenti altri dati di tipo aggregato (es. array di strutture, strutture con array, stringhe ecc.).Approfondimenti: la rappresentazione interna dei vari tipi, la dichiarazione, l'inizializzazione, la lettura e la scrittura, il passaggio ad una funzione ed il suo ritorno, gli impieghi, le relative parole chiave del linguaggio, le funzioni di libreria standard per il trattamento di dati aggregati (es. stringhe).Esempi, esercizi ed applicazioni.						
Unità	U7 - Reference e puntatori	Saper: <ul style="list-style-type: none">Rappresentare, definire e dichiarare reference e puntatori.Scegliere correttamente gli elementi studiati (variabile, reference , puntatore, operatori, parole chiave ecc.) più opportuni per risolvere un determinato problema.Tracciare algoritmi nelle diverse forme, scrivere, compilare, eseguire, effettuare il debug di programmi, che impieghino le parti studiate.				
<ul style="list-style-type: none">Introduzione: reference e puntatori. Impiego nel passaggio dei parametri e nell'elemento di ritorno delle funzioni. Passaggio per valore e per riferimento. Vantaggi e svantaggi.Aritmetica dei puntatori.Puntatori a dati di tipo aggregato.Parametri e valori di ritorno del main.Esempi, esercizi ed applicazioni .						
Unità	U8 - File e stream	Saper: <ul style="list-style-type: none">Aprire, scrivere, leggere, chiudere files e stream ed accedere nelle diverse modalità.Scegliere correttamente gli elementi studiati (tipo di elemento, funzioni, operatori, modalità, tipo di accesso ecc.) più opportuni per risolvere un determinato problema.Saper scrivere e leggere un file di testo e/o binario.				
U6U7U8 LAB	Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati. (*)					

Unità	U9 - Introduzione alla programmazione con LabView						
Prerequisiti	UUPP						
Conoscenze		Abilità \ Competenze		Met. e mezzi	Verifiche	Tempi	Lav. Dom.
<ul style="list-style-type: none">▪ Introduzione all’ambiente LabView: controlli (con gestione Mechanical Action) e indicatori, le palette, gli strumenti, i concetti di connettore ed icona. Gestione degli aspetti grafici (colore, dimensioni, forma, nome, scala ecc degli oggetti, documentazione e tips), etichette ed elementi grafici (linee, rettangoli, ecc).▪ I Virtual Instruments (VI): il VI ed i sub-VI, variabili locali e globali, shared, FGV. Passaggio di dati tra VI distinti. Proprietà dei VI (aspetto, documentazione, revisioni ecc). Gerarchia dei VI. Aprire, chiudere, salvare (eventualmente con opzioni ed in librerie) un VI.▪ Help e Debug: esecuzione passo-passo con visualizzazione del flusso dati, probe, breakpoints. Help: Manuali utente, help, context help, forum e materiali reperibili in rete.▪ I tipi di dato: dati semplici ed aggregati, associabili a variabili, controlli ed indicatori. VI ed operatori sui diversi tipi di dati aggregati e non (es. operazioni logiche, aritmetiche, su vettori e stringhe, ecc).▪ La programmazione: Le strutture fondamentali (for, while, case, sequence ecc.). Tunnel, indexing e shift register. Formula node ed expression node. La gestione del tempo ed i ritardi temporali. Scrittura e lettura di files testuali e binari. Implementazione di macchine a stati.▪ Automati programmati: introduzione al controllo digitale di dispositivi di tipo ON/OFF. Implementazione di macchine a stati. Analisi e sintesi di automi di tipo programmato.▪ Esempi, esercizi ed applicazioni .		<p>Saper:</p> <ul style="list-style-type: none">▪ 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, flowchart o grafo, nel corrispondente programma in LabView e viceversa.▪ Utilizzare la procedura TOP-DOWN e/o BOTTOM-UP strutturata per costruire un algoritmo in LabView.▪ Scegliere correttamente gli elementi studiati più opportuni per risolvere un determinato problema.▪ Leggere, interpretare, comprendere in tutte le sue parti, proprietà ed opzioni di un VI\SubVI.▪ Tracciare in modo completo, mandare in esecuzione nelle diverse modalità, gestire le librerie di un programma in LabView per mezzo dell’elaboratore.▪ Effettuare il debug di un programma.▪ Effettuare una ricerca mediante l’help dell’ambiente ed i manuali.▪ Utilizzare i principali tipi di dati, aggregati e non, le principali funzioni e strumenti dell’ambiente di sviluppo, saperne impostare le opzioni e le proprietà.▪ Operare con blocchi concorrenti nello stesso VI e con VI concorrenti.▪ Prevedere il comportamento di programmi o blocchi di programma▪ Leggere e scrivere files.▪ Realizzare una macchina a stati per risolvere semplici problemi generici e di automazione.▪ Analisi e sintesi e valutazione di semplici sistemi sequenziali programmati, correttamente inizializzati ed interfacciati al mondo esterno (forma simulata).		LF LI LG LM EC LT MA DI AL LA VP TG CD PC AO	VS e/o VO e/o VP	Tot 26 h	ST EX PM
LAB	<ul style="list-style-type: none">▪ Impostazione dell’ambiente di sviluppo.▪ Stesura, esecuzione e debug di programmi in cui sono presenti gli elementi studiati. (*)						

Prerequisiti		UUPP				
Conoscenze		Abilità \ Competenze	Met. e mezzi	Verifiche	Tempi	Lav dom
Unità	U10 - Componenti con memoria	<p>Saper</p> <ul style="list-style-type: none">Leggere, interpretare e comprendere le varie forme con cui possono essere descritti i circuiti sequenziali (grafo, tabella delle transizioni, diagramma temporale, algoritmo/software, circuito elettronico, espressioni logiche, ecc) e saper passare da una all'altra delle rappresentazioni.Descrivere mediante le forme previste il comportamento di un FF o di un dato circuito sequenziale.Utilizzare i dispositivi sequenziali integrati studiati per la realizzazione di circuiti applicativi bassa complessità.Comprendere, interpretare, analizzare e valutare schemi e software che impiegano o realizzano circuiti sequenziali.Comprendere e consultare i manuali e datasheets ed estrarre i parametri principali di un circuito sequenziale.Analisi e sintesi di semplici sistemi sequenziali, cablati e programmati, correttamente iniziati ed interfacciati al mondo esterno.	LF LI LG LM EC LT MA DI AL LA VP TG CD PC AO	VS e/o VO e/o VP	Tot 30h	ST EX PM
<ul style="list-style-type: none">Introduzione ai circuiti sequenziali: definizioni e concetti fondamentali, automi di Mealy e Moore, forme di rappresentazione (grafo, diagramma temporale, tabelle di transizione, descrizione a parole, flowchart, codice, espressioni logiche). Passaggi tra le varie forme. Confronto con i circuiti combinatori. Struttura a blocchi di un automa. Concetti di circuiti asincroni e sincroni.Dispositivi di memoria: Latch e Flip-Flop (FF): generalità, il latch SR, latch SR con enable ed i suoi limiti, il latch D, la logica temporizzata, FF-D, FF-JK, FF-T, level triggered, 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 FFApplicazioni: applicazioni con circuiti sequenziali. Circuito di inizializzazione RC e antirimbolzo. Lettura di datasheets.Esempi, esercizi ed applicazioni						
Unità	U11 - Circuiti sequenziali					
<ul style="list-style-type: none">Dispositivi Sequenziali (DS): contatori sincroni e asincroni (vantaggi e svantaggi) modulo 2^n e modulo qualunque, contatori Up, Down e Up/Down, contatori ad anello, i registri a scorrimento e ad anello, SISO, SIPO, PISO, PIPO uni- e bi-direzionali. Realizzazione mediante FF. Circuiti di conteggio, memorizzazione e visualizzazione ed altre applicazioni. Dispositivi integrati.Automi cablati: Cascata di contatori e diagramma temporale. Contatore come divisore di frequenza. Calcolo della frequenza massima di funzionamento in funzione del tipo di contatore. Analisi e sintesi di automi cablati. Tecnica Drum.Automi programmati: introduzione al controllo digitale di dispositivi di tipo ON/OFF con dispositivi programmabili (es. LabView). Analisi e sintesi di automi di tipo programmato (es. LabView).Esempi, esercizi ed applicazioni						

LAB	<ul style="list-style-type: none"> Esperienze sull'analisi del comportamento e sintesi dei sistemi sequenziali e di semplici applicazioni (es. generatore di sequenza, apricancello, sistemi di automazione in genere, ecc), cablati e programmati, in forma reale e/o simulata. Uso di pacchetti applicativi per la simulazione di circuiti elettronici. Stesura, esecuzione e debug di programmi che implementano circuiti o dispositivi sequenziali. 	<p>Saper</p> <ul style="list-style-type: none"> Effettuare l'analisi, il progetto e la sintesi di sistemi sequenziali correttamente inizializzati ed interfacciati al mondo esterno, cablati e programmati. Realizzazione pratica, collaudo, correzione. Tracciatura del grafo o equivalente. 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) Produrre la documentazione di rito. (*) 				PM
-----	--	---	--	--	--	----

Recupero e ripasso generale	
Ripasso e verifica per il recupero dei contenuti del primo quadrimestre	Da definire
Conclusione dei lavori, ripasso generale della materia, recupero delle insufficienze emerse durante il secondo quadrimestre, approfondimenti	Ore rimanenti

Esperienze pratiche previste

Unità	Esperienze
U3 Algoritmi e analisi dei problemi.	Stesura, verifica del funzionamento e correzione di algoritmi realizzati all'elaboratore con opportuni software di tracciatura e simulazione.
U4 - Introduzione alla programmazione in C\C++	Stesura dell'algoritmo che risolve problemi assegnati contenenti gli elementi studiati. Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati. Compilazione di librerie. Impostazione dell'ambiente e del progetto.
U5 - Procedure e funzioni	Stesura dell'algoritmo che risolve problemi assegnati contenenti gli elementi studiati. Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati. Compilazione di librerie. Impostazione dell'ambiente e del progetto.
U6 - Tipi aggregati U7 - Reference e puntatori U8 - File e stream	Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati.
U9 - Introduzione alla programmazione con LabView	Impostazione dell'ambiente di sviluppo. Stesura, esecuzione e debug di programmi in cui sono presenti gli elementi studiati.
U10 - Componenti con memoria U11 - Circuiti sequenziali	Esperienze sull'analisi del comportamento e sintesi dei sistemi sequenziali e di semplici applicazioni (es. generatore di sequenza, apricancello, sistemi di automazione in genere, ecc), cablati e programmati, in forma reale e/o simulata. Uso di pacchetti applicativi per la simulazione di circuiti elettronici. Stesura, esecuzione e debug di programmi che implementano circuiti o dispositivi sequenziali.

Contenuti irrinunciabili, conoscenze e competenze minime per l'accesso alla classe successiva

Unità	Conoscenze	Abilità \ Competenze
U1 I numeri ed i codici	<ul style="list-style-type: none"> ▪ Sistemi 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. Codifica in virgola fissa. Overflow e Underflow nelle operazioni con e senza segno. Bit segnalatori CNZV ▪ Numeri relativi: modulo e segno e complemento a due. Operazioni. ▪ Codici: ASCII esteso e con bit di parità, Gray e BCD. Proprietà ed impieghi dei diversi i codici. 	<p>Saper</p> <ul style="list-style-type: none"> ▪ 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 con i numeri con segno. ▪ Riconoscere le situazioni di overflow e underflow nelle operazioni a lunghezza fissa con o senza segno. Impostare i segnalatori CNZV. ▪ Convertire un numero/simbolo come previsto dalle specifiche dei diversi codici e viceversa.
U2 Il computer ed il software	<ul style="list-style-type: none"> ▪ Configurazione fondamentale di un PC, hardware e software. Lo schema a blocchi ed il suo funzionamento, la struttura hardware e software di un P.C. e le funzioni delle singole parti e le relazioni reciproche. Sequenza di boot e shutdown. Interprete e compilatore, linker, editor, debugger. Funzioni e caratteristiche. ▪ Livelli di descrizione di un sistema del funzionamento e di funzionalità di un PC. ▪ Livelli dei linguaggi di programmazione. Ambiente di sviluppo. 	<p>Saper</p> <ul style="list-style-type: none"> ▪ Riprodurre la struttura per schema a blocchi di un elaboratore (aspetti hardware e software) e saperne descrivere le caratteristiche, la funzione svolta nell'insieme, da ciascuna parte e le reciproche relazioni intercorrenti. ▪ Interpretare correttamente la sequenza delle fasi di accensione e spegnimento di un P.C., avvio e chiusura di una applicazione. ▪ Riconoscere vantaggi e svantaggi nell'uso di un compilatore o interprete ed i migliori campi di impiego. ▪ Saper descrivere le funzioni specifiche di produzione del software ed i vantaggi e svantaggi dei software di diverso livello.
U3 Algoritmi e analisi dei problemi.	<ul style="list-style-type: none"> ▪ Concetti di algoritmo e programma. ▪ Le tecniche di rappresentazione mediante flowchart e pseudocodice. ▪ I blocchi e le strutture di controllo, gli algoritmi strutturati e non e le loro proprietà. Proprietà. ▪ Criteri di progetto ed ottimizzazione di un algoritmo. ▪ Laboratorio: Stesura, verifica del funzionamento e correzione di algoritmi realizzati all'elaboratore con opportuni software di tracciatura e simulazione.(*) 	<p>Saper:</p> <ul style="list-style-type: none"> ▪ Analisi, sintesi e valutazione di un algoritmo in una delle forme studiate, relativamente a semplici problemi utilizzando tecniche, metodi, strumenti, criteri studiati. ▪ Riconoscere, interpretare ed utilizzare il linguaggio e la simbologia impiegata per descrivere un algoritmo. Comprendere un assegnato algoritmo rappresentato mediante le forme studiate e saperlo rappresentare in una qualsiasi altra forma. ▪ Riconoscere se un algoritmo è dato in forma strutturata o meno e saperlo strutturare. ▪ Saper identificare i blocchi e le strutture studiate in un semplice algoritmo fornito in una qualsiasi forma.

U4 Introduzione e alla programmazione in C	<ul style="list-style-type: none"> I concetti base del linguaggio C/C++, la stampa e l'input formattato e non, le direttive, le variabili e le costanti, gli identificatori, le parole chiave. I tipi fondamentali, le dichiarazioni e definizioni, le espressioni, gli operatori. Precedenze e associatività. Le conversioni. Le regole per la stesura del software. I commenti. Enumerativi. Vita e visibilità, variabili locali e globali, Le strutture di programmazione fondamentali. Concetti base di programmazione, programmi batch e ciclici, le variabili interruttore e di accumulo. Editor, compilatore, linker, debugger. I codice sorgente, oggetto ed eseguibile, le librerie. Impiego dei manuali e dell'Help d'ambiente e on-line. Errori logici e sintattici. 	<p>Saper:</p> <ul style="list-style-type: none"> Scegliere correttamente gli elementi studiati più opportuni per risolvere un determinato problema. Effettuare l'analisi, la sintesi e la valutazione di software scritti in C/C++ relativi a semplici problemi contenenti le parti studiate. Tradurre un semplice algoritmo dato in pseudocodice o flowchart nel corrispondente programma in linguaggio C/C++ e viceversa. Utilizzare l'ambiente di sviluppo per creare il progetto, editare, commentare il codice, compilare, effettuare il debug ed eseguire semplici programmi in C/C++. Effettuare una ricerca mediante l'help online e dell'ambiente ed i manuali.
U5 Procedure e funzioni	<ul style="list-style-type: none"> Dichiarazione, definizione, chiamata ed esecuzione di una funzione. Struttura di un programma con le funzioni. Vantaggi e svantaggi nell'uso delle funzioni. Flowchart e pseudocodice di algoritmi con funzioni. Parametri e valore di ritorno di una funzione. Variabili globali e locali. I prototipi, il preprocessore e le direttive di compilazione. Periodo di vita e visibilità di variabili e funzioni. Dichiarazioni nei diversi livelli. Inizializzazione e classi di memorizzazione. Overloading di funzioni, ridefinizione di variabili. File sorgenti e di intestazione. Le librerie. Namespace e Using, Scope resolution. 	
U4U5 LAB	<ul style="list-style-type: none"> Stesura dell'algoritmo che risolve semplici problemi. Stesura del codice con adeguati commenti, link, compilazione, esecuzione, debug del relativo codice in linguaggio C/C++. Compilazione di librerie. Impostazione dell'ambiente e del progetto 	
U6 Tipi aggregati	<ul style="list-style-type: none"> Array e matrici, Strutture. Operazioni con dati aggregati. Vettori di caratteri e stringhe. Dichiarazione di tipi. Dati di tipo aggregato contenenti altri dati di tipo aggregato. Dichiarazione, inizializzazione, lettura e scrittura, passaggio ad una funzione ed il suo ritorno, gli impieghi, le relative parole chiave del linguaggio, le funzioni di libreria standard per il trattamento di dati aggregati (es. stringhe). 	<p>Saper:</p> <ul style="list-style-type: none"> Analisi, sintesi e valutazione di semplici algoritmi contenenti gli elementi studiati. Rappresentare, definire e dichiarare i tipi studiati, reference e puntatori. Scegliere correttamente gli elementi studiati più opportuni per risolvere un semplice problema. Tracciare algoritmi nelle diverse forme, scrivere, compilare, eseguire, effettuare il debug di semplici programmi, che impieghino le parti studiate.
U7 Reference e puntatori	<ul style="list-style-type: none"> Reference e puntatori. Impiego nel passaggio dei parametri e nell'elemento di ritorno delle funzioni. Passaggio per valore e per riferimento. Vantaggi e svantaggi. Aritmetica dei puntatori, puntatori a dati di tipo aggregato, parametri e valori di ritorno del main. 	
U8 File e stream	<ul style="list-style-type: none"> Files e flussi di testo e binari. Accesso, apertura, scrittura, lettura, chiusura di un file. Stream di input/output, input e output non formattati, accessi sequenziali e random, stream come argomenti di funzioni. Svuotamento di uno stream ed eliminazione di un file. 	
U6U7U8 LAB	Scrittura, compilazione, link e debug di programmi in cui sono presenti gli elementi studiati.(*)	

<p>U9 Introduzione alla programmazione con LabView</p>	<ul style="list-style-type: none"> ▪ L'ambiente LabView: controlli (con gestione Mechanical Action) e indicatori, le palette, gli strumenti, i concetti di connettore ed icona. Gestione degli aspetti grafici (colore, dimensioni, forma, nome, scala ecc degli oggetti, documentazione e tips), etichette ed elementi grafici (linee, rettangoli, ecc). ▪ I Virtual Instruments (VI) ed i sub-VI, variabili locali e globali, shared, FGV. Passaggio di dati tra VI distinti. Proprietà dei VI (aspetto, documentazione, revisioni ecc). Gerarchia dei VI. Aprire, chiudere, salvare (eventualmente con opzioni ed in librerie) un VI. ▪ Debug, esecuzione passo-passo con visualizzazione del flusso dati, probe, breakpoints. Help, manuali utente, help, context help, forum e materiali reperibili in rete. ▪ I tipi di dato, dati semplici ed aggregati, associabili a variabili, controlli ed indicatori. VI ed operatori sui diversi tipi di dati aggregati e non (es. operazioni logiche, aritmetiche, su vettori e stringhe, ecc). ▪ La programmazione. le strutture fondamentali. Tunnel, indexing e shift register. Formula node ed expression node. La gestione del tempo ed i ritardi temporali. ▪ Scrittura e lettura di files testuali e binari. Implementazione di macchine a stati. <p>Laboratorio: impiego dello strumento di sviluppo per la realizzazione di software che impiegano gli elementi studiati. Impostazione dell'ambiente di sviluppo</p>	<p>Saper:</p> <ul style="list-style-type: none"> ▪ Riconoscere, scegliere, e saper utilizzare e impostare correttamente tutti gli elementi studiati nei programmi al fine di ottenere il risultato voluto. ▪ 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 flowchart nel corrispondente programma in LabView e viceversa. ▪ Tracciare in modo completo, mandare in esecuzione nelle diverse modalità, gestire le librerie di un programma in LabView per mezzo dell'elaboratore. ▪ Effettuare una ricerca mediante l'help dell'ambiente ed i manuali. ▪ Descrivere ed utilizzare le principali funzioni e strumenti dell'ambiente di sviluppo, dei VI e di suoi elementi, saperne impostare le opzioni e le proprietà. ▪ Operare con blocchi concorrenti nello stesso VI e con VI concorrenti. ▪ Prevedere il comportamento di programmi o blocchi di programma ▪ Analisi, sintesi e valutazione di programmi in LabView contenenti gli elementi studiati per risolvere semplici problemi di tipo generico e di automazione, sia con tecnica generale sia mediante automi a stati finiti. ▪ Editare, eseguire, effettuare il debug di programmi in cui sono presenti gli elementi studiati. (*)
<p>U10 Componenti con memoria</p>	<ul style="list-style-type: none"> ▪ I circuiti sequenziali, definizioni e concetti fondamentali, automi di Mealy e Moore, forme di rappresentazione. Passaggi tra le varie forme. Struttura di un automa. ▪ Dispositivi di memoria: Latch e Flip-Flop (FF), latch D, FF-D, FF-JK, FF-T, level triggered, positive/negative edge-triggered e Master-Slave. Descrizioni del funzionamento mediante grafo e tabelle di funzionamento, i tipici segnali di controllo, ingressi diretti asincroni, caratteristiche temporali dei FF ▪ Applicazioni: semplici applicazioni con circuiti sequenziali. Circuito di inizializzazione RC e antirimbando. Lettura di datasheets. 	<p>Saper</p> <ul style="list-style-type: none"> ▪ Leggere, interpretare e comprendere le varie forme con cui possono essere descritti i circuiti sequenziali (grafo, tabella delle transizioni, diagramma temporale, software, circuito elettronico, espressioni logiche, ecc) e saper passare da una all'altra delle rappresentazioni. ▪ Descrivere mediante le forme previste il comportamento di un FF o di un dato circuito sequenziale.
<p>U11 Circuiti sequenziali</p>	<ul style="list-style-type: none"> ▪ Dispositivi Sequenziali. Contatori sincroni e asincroni, Up, Down e Up/Down, contatori ad anello (vantaggi e svantaggi) modulo qualunque, i registri: a scorrimento e ad anello, SISO, SIPO, PISO, PIPO uni- e bi-direzionali, Realizzazione con FF. Circuiti di conteggio, memorizzazione e visualizzazione ed altre semplici applicazioni. Dispositivi integrati. ▪ Automi cablati. Cascata di contatori e diagramma temporale. Contatore come divisore di frequenza. Calcolo della frequenza massima di funzionamento in funzione del tipo di contatore. Analisi e sintesi di semplici automi cablati. Tecnica Drum. ▪ Automi programmati, controllo digitale di semplici dispositivi di tipo ON/OFF con dispositivi programmabili attraverso le operazioni su singoli bit. Analisi e sintesi di automi di tipo programmato. 	<ul style="list-style-type: none"> ▪ Analisi, sintesi, valutazione di semplici circuiti applicativi che impiegano i dispositivi sequenziali studiati. ▪ Consultare e comprendere i manuali e datasheets ed estrarre i parametri principali di un circuito sequenziale. ▪ Analisi e sintesi di semplici sistemi sequenziali, cablati e programmati, correttamente inizializzati ed interfacciati al mondo esterno.

NOTE

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

1° **Prerequisiti:**

UUPP Unità didattiche precedenti e relativi prerequisiti

4° **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
TG Tavola grafica
CD Materiali in formato elettronico (CD-Rom, pagine web, ecc.)
PC Personal Computer, relativi pacchetti applicativi, internet.
AO Attrezzatura ordinaria del laboratorio

5° **Verifiche:****Teorico:**

VS Verifica scritta
VO Verifica orale

Grafico:

VG Verifica grafica

Pratico:

VP Verifica pratica

6° **Tempi:**

V Verifica scritta
T Teoria
P Laboratorio, Pratica (Esercitazione)
R-R Recupero, Ripasso

7° **Lavoro domestico:**

ST Studio teorico
EX Esercizi
PM Produzione di materiali: es. schemi, relazioni, disegni, elaborazione dati, simulazioni, software, ecc.

...

Lencenigo, 17 / 10 / 2024

I docenti del Dipartimento