IIS CIGNA - BARUFFI - GARELLI
MONDOVÌ

PROGRAMMAZIONE ANNUALE
ANNO SCOLASTICO 2025/26

INSEGNAMENTO	**Sistemi Automatici**
DOCENTE	**Bertolino Davide, Gasco Giovanni**
CLASSE	**3AEE**
INDIRIZZO	**Elettronica ed Elettrotecnica**

METODO DI INSEGNAMENTO

- Lezione frontale e dialogata
- Problem solving e coding guidato
- Esercitazioni pratiche su simulatore (Tinkercad) e su hardware reale (Arduino)
- Lavori di gruppo e peer programming
- Laboratorio ed esercitazioni

STRUMENTI DI LAVORO

- LIM / Lavagna classica
- Dispense e appunti del docente
- Piattaforma Tinkercad Circuits (Autodesk)
- Schede Arduino Uno/Nano e componenti elettronici
- IDE Arduino e Monitor Seriale
- Software di simulazione circuitale

STRUMENTI DI VERIFICA

- Prove scritte e test strutturati
- Interrogazioni orali
- Relazioni di laboratorio
- Valutazione di progetti e prototipi funzionanti

OBIETTIVI

- Comprendere l'architettura di un sistema a microcontrollore e le sue periferiche fondamentali

- Sviluppare programmi in C/C++ per Arduino con complessità crescente
- Progettare e realizzare interfacce CLI per l'interazione uomo-macchina via porta seriale
- Conoscere e utilizzare le tipologie di memoria di un microcontrollore (Flash, SRAM, EEPROM)
- Gestire la comunicazione seriale (UART) per il monitoraggio e il controllo di sistemi embedded
- Applicare un approccio metodico alla risoluzione di problemi di automazione

CONTENUTI DISCIPLINARI

UDA/MOD.1 — INTRODUZIONE AD ARDUINO E ALL'AMBIENTE DI SVILUPPO

- Il microcontrollore ATmega328P: architettura interna e pinout
- La scheda Arduino Uno: alimentazione, pin digitali, pin analogici, pin PWM
- L'IDE Arduino: struttura di uno sketch (setup, loop), compilazione e upload
- Tinkercad Circuits: introduzione all'ambiente di simulazione
- Tipi di dati fondamentali in C/C++ per Arduino: int, float, char, bool, byte
- Variabili, costanti e direttive #define
- Strutture di controllo: if/else, switch/case, for, while, do/while
- Funzioni: dichiarazione, parametri, valori di ritorno
- Il concetto di libreria e l'inclusione con #include

UDA/MOD.2 — INPUT/OUTPUT DIGITALE E ANALOGICO

- Configurazione dei pin: pinMode(), digitalWrite(), digitalRead()
- Gestione di LED, pulsanti e buzzer
- Resistenze di pull-up e pull-down: funzione e dimensionamento
- Il pull-up interno di Arduino: INPUT_PULLUP
- Anti-rimbalzo (debounce) software dei pulsanti
- Lettura analogica: analogRead() e il convertitore ADC a 10 bit
- Uscita PWM: analogWrite() e il concetto di duty cycle
- Controllo di luminosità LED e velocità motori DC tramite PWM
- Lettura di sensori analogici: potenziometri, fotoresistenze (LDR), sensori di temperatura (NTC/LM35)
- Funzione map() per la conversione tra range di valori

UDA/MOD.3 — COMUNICAZIONE SERIALE E INTERFACCIA CLI

- La porta seriale UART: principio di funzionamento, TX, RX, baud rate
- Inizializzazione: Serial.begin() e scelta del baud rate
- Invio di dati al PC: Serial.print(), Serial.println(), Serial.write()
- Ricezione di dati dal PC: Serial.available(), Serial.read(), Serial.readString()
- Il Monitor Seriale dell'IDE Arduino: utilizzo per debug e test
- Progettazione di un'interfaccia CLI (Command Line Interface): concetto e architettura
- Parsing dei comandi: tokenizzazione di stringhe, confronto con strcmp()
- Implementazione di un menu interattivo via seriale
- Comandi con parametri: parsing di valori numerici da stringa
- Gestione degli errori di input e feedback all'utente
- Protocolli di comunicazione seriale: cenni su I²C e SPI

UDA/MOD.4 — GESTIONE DELLA MEMORIA IN ARDUINO

- Le tre aree di memoria del microcontrollore: Flash, SRAM, EEPROM
- Memoria Flash: spazio programma e limiti (32 KB su ATmega328P)
- Memoria SRAM: variabili, stack e heap — limiti (2 KB su ATmega328P)
- La macro F() per lo spostamento delle stringhe costanti in Flash
- PROGMEM: memorizzazione di dati costanti in Flash
- Memoria EEPROM: caratteristiche, limiti di scrittura, persistenza dei dati
- Libreria EEPROM: EEPROM.read(), EEPROM.write(), EEPROM.update()
- EEPROM.put() e EEPROM.get() per strutture dati complesse

- Ottimizzazione dell'uso della memoria: tecniche e buone pratiche
- Diagnostica: verifica dello spazio libero in SRAM a runtime

UDA/MOD.5 — PROBLEMI APPLICATIVI E PROGETTI

- Approccio metodico alla risoluzione di problemi: analisi, progettazione, codifica, test
- Semaforo intelligente: gestione temporizzata di LED con logica a stati
- Termometro digitale con visualizzazione su Monitor Seriale
- Sistema di allarme con sensori e soglie configurabili via CLI
- Data logger su EEPROM con lettura successiva via seriale
- Controllo di un servomotore tramite comandi seriali
- Automazione di un processo semplice: macchina a stati finiti
- Cenni sulla documentazione tecnica di un progetto embedded

ATTIVITÀ DI LABORATORIO

- Primo approccio a Tinkercad Circuits: simulazione di un circuito LED + resistenza
- Programmazione base: accensione/spegnimento LED, sequenze luminose (Knight Rider)
- Lettura di pulsanti con debounce e controllo di attuatori
- Lettura di sensori analogici (potenziometro, LDR, NTC) e visualizzazione su Monitor Seriale
- Controllo PWM di LED e motori DC con potenziometro
- Sviluppo di un'interfaccia CLI: menu interattivo con comandi testuali
- CLI avanzata: comandi con parametri per il controllo di LED e attuatori
- Esplorazione della memoria: verifica SRAM libera, uso della macro F()
- Scrittura e lettura di configurazioni persistenti su EEPROM
- Realizzazione di un data logger con salvataggio su EEPROM e scaricamento via seriale
- Progetto semaforo intelligente con logica a stati finiti
- Progetto integrato: sistema di monitoraggio con CLI, sensori e EEPROM

In fede,

I Docenti: **Bertolino Davide, Gasco Giovanni**