

IIS CIGNA-BARUFFI-GARELLI — MONDOVÌ

PROGRAMMA SVOLTO

Anno Scolastico 2025/26

Insegnamento	Sistemi Automatici
Docenti	Bertolino Davide, Gasco Giovanni
Classe	4AEE
Indirizzo	Elettronica ed Elettrotecnica

Obiettivi

- Consolidare le competenze di programmazione su piattaforma Arduino con periferiche avanzate
- Pilotare display grafici (OLED, LCD) per la visualizzazione di dati e interfacce utente
- Conoscere l'architettura dei microcontrollori PIC (PIC16F84A e PIC16F887) e la loro programmazione in Assembly
- Comprendere il modello degli automi a stati finiti (FSM) e applicarlo alla progettazione di sistemi di controllo
- Sviluppare semplici programmi in linguaggio Assembly PIC per la gestione di I/O e temporizzazioni
- Tradurre specifiche funzionali in diagrammi a stati e successivamente in codice eseguibile

Contenuti disciplinari svolti

UDA/MOD. 1 — Ripasso e approfondimento Arduino

- Ripasso dell'architettura Arduino e del ciclo di sviluppo: sketch, compilazione, upload
- Ripasso delle strutture di controllo, funzioni e gestione I/O digitale e analogico
- Comunicazione seriale UART: ripasso di Serial.print(), Serial.read() e interfacce CLI
- Gestione della memoria: ripasso di Flash, SRAM, EEPROM e macro F()
- Array, strutture (struct) e puntatori in C/C++ per Arduino
- Uso di librerie esterne: installazione, inclusione e configurazione
- Timer e interrupt: concetti fondamentali e applicazioni con attachInterrupt()
- Millis() vs delay(): programmazione non bloccante e multitasking cooperativo

UDA/MOD. 2 — Display grafici e interfacce utente

- Display LCD 16x2 con interfaccia I²C: inizializzazione e libreria LiquidCrystal_I2C
- Visualizzazione di testo, variabili e caratteri personalizzati su LCD
- Il bus I²C: principio di funzionamento, indirizzamento, SDA e SCL
- Display OLED SSD1306 (128x64) con interfaccia I²C: libreria Adafruit_SSD1306
- Primitive grafiche su OLED: testo, linee, rettangoli, cerchi, bitmap
- Progettazione di interfacce utente grafiche: menu, barre di progresso, grafici real-time
- Navigazione nei menu tramite pulsanti e encoder rotativo
- Visualizzazione di dati da sensori su display grafico

UDA/MOD. 3 — Architettura dei microcontrollori PIC

- Introduzione alla famiglia Microchip PIC: panoramica e classificazione
- Architettura Harvard vs Von Neumann: differenze e implicazioni
- Il PIC16F84A: architettura interna, pinout, organizzazione della memoria (Program, Data, EEPROM)
- Registri fondamentali: W, STATUS, PORTA, PORTB, TRISA, TRISB
- Il PIC16F887: architettura avanzata, pinout esteso, periferiche integrate
- Organizzazione della memoria del PIC16F887: banchi di registro, SFR e GPR
- Modulo ADC del PIC16F887: configurazione e utilizzo

- Timer del PIC: Timer0, prescaler e generazione di temporizzazioni
- L'ambiente di sviluppo MPLAB X IDE: creazione di un progetto, configurazione del dispositivo
- Il programmatore PICKit: connessione e procedura di programmazione

UDA/MOD. 4 — Programmazione Assembly PIC

- Introduzione al linguaggio Assembly: mnemonici, operandi, commenti
- Set di istruzioni del PIC16F: istruzioni byte-oriented, bit-oriented e literal/control
- Istruzioni fondamentali: MOVLW, MOVWF, MOVF, CLRF, BSF, BCF, BTFSC, BTFSS
- Istruzioni aritmetiche e logiche: ADDLW, ADDWF, SUBLW, SUBWF, ANDLW, IORLW, XORLW
- Istruzioni di salto e controllo: GOTO, CALL, RETURN, RETLW
- Direttive assembler: ORG, EQU, #include, __CONFIG, END
- Configurazione delle porte I/O: registri TRISx e PORTx
- Gestione dell'accensione e spegnimento di LED tramite Assembly
- Cicli di ritardo software: calcolo dei tempi con cicli annidati
- Lettura di pulsanti e gestione dell'anti-rimbalzo in Assembly
- Subroutine: organizzazione modulare del codice Assembly
- Tabelle di lookup con RETLW per conversione dati

UDA/MOD. 5 — Automi a stati finiti (FSM)

- Concetto di sistema a stati: stato, transizione, evento, azione
- Automi a stati finiti (FSM): definizione formale e componenti
- Rappresentazione grafica: diagrammi a stati (state diagram)
- Tabelle di transizione degli stati
- Traduzione di specifiche funzionali in diagrammi a stati
- Implementazione di FSM in C/C++ su Arduino: approccio con switch/case
- Implementazione di FSM in Assembly PIC: uso di variabili di stato e salti condizionati
- Esempi applicativi: semaforo, distributore automatico, sistema di accesso con codice
- FSM temporizzate: gestione dei timeout e delle transizioni a tempo
- Validazione e test di un automa: verifica della copertura degli stati

Attività di laboratorio

- Ripasso Arduino: esercizi su I/O, comunicazione seriale e CLI
- Collegamento e pilotaggio di un display LCD 16x2 via I²C
- Creazione di caratteri personalizzati su LCD
- Collegamento e pilotaggio di un display OLED SSD1306 via I²C
- Realizzazione di un'interfaccia utente grafica con menu navigabile su OLED
- Visualizzazione real-time di dati da sensore di temperatura su display grafico
- Primo progetto MPLAB X: LED lampeggiante in Assembly su PIC16F84A (simulazione)
- Programma Assembly: sequenza luminosa su PORTB del PIC16F84A
- Programma Assembly: lettura di un pulsante e controllo di un LED con anti-rimbalzo
- Programma Assembly: cicli di ritardo calibrati e generazione di temporizzazioni
- Implementazione di un semaforo come FSM su Arduino con display di stato su OLED
- Implementazione di un automa FSM in Assembly PIC: distributore automatico semplificato
- Progetto integrato: sistema di controllo accessi con FSM, tastiera e display

Mondovì, 10 / 06 / 2026

In fede, i Docenti:

Bertolino Davide _____

Gasco Giovanni _____