

Programma degli argomenti svolti

anno scolastico: **2018/2019**
classe: **3F (ITT – articolazione Telecomunicazioni)**
materia: **Tecnologia e Progettazione di Sistemi Informatici e di Telecomunicazioni**
docenti: **Prof. Lorenzo Donati, Prof. Mario D'Antoni**

Nota: il presente programma non riflette pedissequamente la distribuzione temporale degli argomenti durante l'anno scolastico, bensì è organizzato secondo blocchi tematici omogenei. In grassetto sono indicati gli argomenti indispensabili per il raggiungimento degli obiettivi didattici minimi; su tali argomenti si baserà la prova di esame per gli alunni con giudizio sospeso.

□ **Elettronica Digitale**

○ **Algebra di Boole**

- ◆ Stati logici elementari e bit, parole binarie, variabili logiche.
- ◆ Operazioni elementari AND, OR e NOT. Espressioni logiche. Espressioni logiche equivalenti ed identità logiche. Determinazione della tabella di verità di un'espressione logica.
- ◆ Proprietà dell'algebra di Boole: associativa, commutativa, distributiva, doppia negazione, dei complementi, di idempotenza, di neutralità, di annullamento, di DeMorgan, di assorbimento, di assorbimento del complemento.
- ◆ Operazioni bit a bit tra parole binarie.

○ **Reti logiche**

- ◆ Porte logiche: AND, OR, NOT, buffer, NAND, NOR, XOR, XNOR. Circuiti logici. Circuiti logici equivalenti. Porte a più ingressi.
- ◆ Universalità delle porte NAND e NOR. Porte 3-state e cenni alle applicazioni (bus bidirezionali, bus condivisi).
- ◆ Determinazione della legge di commutazione di una rete combinatoria a partire dal suo schema logico e viceversa.
- ◆ Concetto di livello logico attivo; ingressi attivi alti ed attivi bassi.
- ◆ Multiplexer (MUX), demultiplexer (DEMUX) e decoder binari. Applicazioni.
- ◆ Riconoscimento di stati logici. Mintermini. Forma canonica SP e relativo circuito canonico.
- ◆ Diagrammi di temporizzazione. Ritardo di propagazione nelle reti logiche.
- ◆ Reti sequenziali sincrone ed asincrone: latch SR, latch D, flip-flop D.
- ◆ Pilotaggio delle reti sequenziali: tempo di setup e tempo di hold.
- ◆ Applicazioni dei flip-flop e dei latch: registri di memoria, registri a scorrimento, contatori, divisori di frequenza.

○ **Tecnologia dei circuiti digitali**

- ◆ Rappresentazione dei livelli logici con livelli di tensione elettrica.
- ◆ Tecnologie digitali attuali: CMOS ed ECL, differenze principali e loro ambiti applicativi. Cenni alla tecnologia obsoleta TTL e legami con le tecnologie attuali. Famiglie logiche CMOS e compatibilità con le TTL.
- ◆ Cenni ai transistori MOSFET: funzionamento ON/OFF semplificato.
- ◆ Struttura e funzionamento di un inverter CMOS. Implementazione delle uscite 3-state mediante struttura CMOS.

- ◆ Porte di trasmissione CMOS e loro applicazioni (in particolare: realizzazione di MUX/DEMUX analogici e porte 3-state).
- ◆ Esempio concreto di applicazione dei dispositivi studiati (porte 3-state, registri a scorrimento, registri latch): registro seriale-parallelo 74HC595, lettura del suo datasheet ed esame dello schema logico interno.
- ◆ Bus seriali e bus paralleli. Cenni ai bus seriali standard UART, SPI, I2C e CANbus.
- ◆ Cenni all'interfaccia JTAG per il debugging/programmazione di sistemi con MCU/FPGA.
- ◆ Cenni alla conversione seriale-parallelo usando l'integrato 74HC595.

○ **Tecniche di interfacciamento dei circuiti digitali**

- ◆ Cenni al funzionamento dei diodi a semiconduttore (diodi raddrizzatori, diodi Zener e LED).
- ◆ Cenni al funzionamento dei regolatori di tensione integrati.
- ◆ Cenni al funzionamento dei comparatori di tensione analogici.
- ◆ Pilotaggio di un LED mediante un'uscita digitale. Dimensionamento della resistenza di limitazione.
- ◆ Contatti elettrici normalmente chiusi e normalmente aperti e loro uso come sorgenti di livelli logici. Resistori di pull-up e di pull-down.
- ◆ Il problema dei rimbalzi durante l'azionamento di contatti elettrici e caratteristiche del fenomeno.
- ◆ Segnali PWM e pilotaggio di carichi mediante tecnica PWM.

○ **Microcontrollore ATmega328P**

- ◆ Schema a blocchi della MCU e funzionalità di base dei singoli blocchi.
- ◆ Caratteristiche hardware e software principali della MCU (memorie RAM, EEPROM e flash; architettura di Harvard).
- ◆ Struttura e funzionamento delle porte GPIO (General Purpose Input/Output). Configurazione e utilizzo delle porte GPIO.
- ◆ Sistema di gestione dell'alimentazione e del reset.
- ◆ Le interruzioni nel microcontrollore ATmega328P: sorgenti di interruzione ed eventi collegati.
- ◆ Sistema Watchdog, sue caratteristiche e suo utilizzo.
- ◆ Unità timer/contatori: struttura interna, funzionalità e configurazione.
- ◆ Cenni al funzionamento del blocco comparatore analogico ed al riferimento "bandgap".
- ◆ Interfaccia USART: caratteristiche generali, caratteristiche del protocollo di comunicazione in modalità UART, caratteristiche fisiche del collegamento. Parametri di comunicazione in modalità UART e configurazione dell'interfaccia.

- ◆ Cenni al blocco convertitore analogico/digitale.
- ◆ Cenni ai blocchi di interfaccia I2C ed SPI.

○ Scheda Arduino UNO

- ◆ Analisi dello schema elettrico della scheda Arduino UNO basata su MCU ATmega328P.
- ◆ Blocco convertitore da porta UART a porta USB per il collegamento al PC basato su MCU ATmega16U2.
- ◆ Circuiti di generazione del clock (quarzi con condensatori di carico), circuiti di protezione (varistori, induttanze e fusibile autoripristinante) e circuito di reset.
- ◆ Circuito di pilotaggio del LED sul pin 13 della scheda.
- ◆ Circuito di alimentazione con sistema di commutazione automatica da alimentazione esterna ad alimentazione da porta USB.

□ Programmazione dei microcontrollori

○ Fondamenti

- ◆ Cenni sulle Macchine a stati finiti.
- ◆ Programmazione di sistemi embedded: problematiche e particolarità. Sistemi in tempo reale soft e hard. Sistemi embedded basati su microcontrollore (MCU).
- ◆ Cenni ad altri dispositivi usati nei sistemi embedded (DSP, microprocessori, FPGA).
- ◆ Elementi di programmazione concorrente: processi, task, parallelismo reale e virtuale, scheduling collaborativo (collaborative/non-preemptive scheduling) e con prerilascio (preemptive scheduling). Risorse condivise e sezioni critiche del codice.
- ◆ Lo stack come struttura dati. Stack hardware di un microprocessore e suoi utilizzi. Cenni alle chiamate a subroutine assembly (Istruzioni assembly JSR e RET).
- ◆ Il meccanismo delle interruzioni. Routine di gestione delle interruzioni (ISR – Interrupt Service Routine) e vettore delle interruzioni (Istruzione assembly IRET).

- ◆ Abilitazione/disabilitazione globale delle interruzioni (istruzioni assembly CLI e SEI). Mascheramento delle interruzioni.
- ◆ Etica e sicurezza nell'ambito delle applicazioni embedded. Principi di ingegneria del software per la realizzazione di sistemi affidabili. Analisi degli incidenti mortali che recentemente hanno coinvolto gli aerei modello Boeing 737 Max.

○ Programmazione in linguaggio C della scheda Arduino UNO

- ◆ Fasi della compilazione in un programma in linguaggio C/C++.
- ◆ Preprocessore C: macrooggetti (object-like macros) e macrofunzioni (function-like macros); direttiva #define e direttiva #undef. Uso delle macro nella programmazione embedded.
- ◆ Linguaggio C: tipi enumerati (enum) e istruzione switch; operazioni bit a bit e operazioni di shift (operatori |, &, ~, ^, <<, >>). Chiamata e definizione di funzioni.
- ◆ Programmazione modulare in C/C+: suddivisione del programma in file sorgente multipli.
- ◆ Bootloader ed uso dell'interfaccia UART per la programmazione "in-circuit".
- ◆ Uso dell'ambiente Arduino per la programmazione della scheda Arduino UNO. Struttura generale di un programma Arduino: funzioni setup, loop, pinMode, digitalWrite, delay). Funzionalità di base delle librerie di Arduino ed alternative standard in linguaggio C a basso livello.
- ◆ Uso delle porte GPIO per pilotare carichi esterni e per leggere lo stato di contatti elettromeccanici.
- ◆ Implementazione firmware di una macchina a stati finiti antirimbato per la lettura di ingressi digitali pilotati da contatti elettromeccanici.
- ◆ Uso dei timer per la generazione hardware di onde quadre e segnali PWM.
- ◆ Definizione delle routine di gestione delle interruzioni in C mediante libreria AVR-libc.
- ◆ Implementazione di un semplice algoritmo di scheduling collaborativo per sistemi soft real-time.