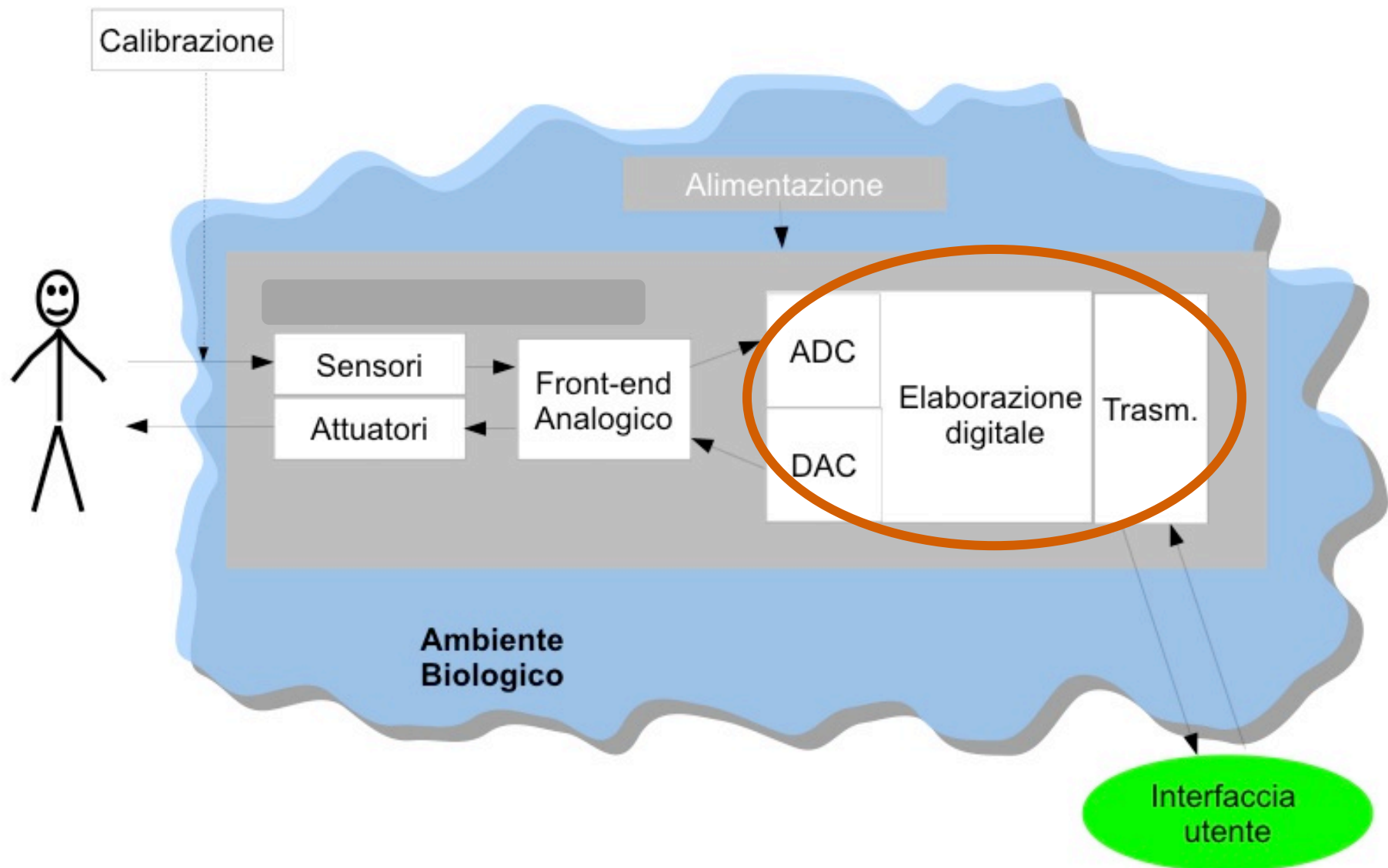


Misure in Campo Biomedico

Elaborazione dati di sensori: la scheda Arduino

Schema Generale sistema di misura biomedico



Sottosistemi

Nell'ambito della progettazione di sistemi di misura l'Ingegnere Biomedico dovrà essere in grado di conoscere le principali caratteristiche/problematiche relative ai singoli sottoinsiemi che costituiscono il dispositivo.

▪ Sensori

- Costituiscono la principale interfaccia con il paziente servono per misurare le variabili di interesse
 - bioelettrodi, fisici (temperatura, deformazione, velocità/accelerazione), ottici, chimici/elettrochimici

▪ Attuatori

- si interfacciano al paziente fornendo l'”energia” necessaria
 - per terapia
 - per valutare l'interazione dell'energia trasmessa ai fini della misura di parametri di interesse diagnostico

Sottosistemi

▪ Front-end analogico

- Anche se gran parte delle operazioni più complesse potrà essere fatta in digitale, la circuitistica analogica di interfacciamento ai sensori/attuatori è una componente essenziale e imprescindibile di un dispositivo biomedicale
 - amplificatori per la trattazione di segnali analogici provenienti dai sensori (amplificatori da strumentazione)
 - filtraggio passa basso anti-aliasing prima del blocco di conversione analogica/digitale
 - amplificatori da isolamento per problematiche di sicurezza legate alla circuitistica di alimentazione
 - circuitistica di pilotaggio degli attuatori

Sottosistemi

▪ Conversione A/D

- Il segnale analogico è un segnale **continuo a tempo continuo** che contiene un'informazione **limitata**
- **Conversione A/D**: trasferire l'informazione contenuta nel segnale analogico in una successione di numeri interi rappresentati con una serie di simboli di un alfabeto finito.
 - **Campionamento**: passaggio dal segnale continuo a una successione di campioni ($v(t) \rightarrow v(kT)$ $k=0,1,2,\dots$)
 - **Quantizzazione**: ad ogni campione viene associato un numero intero rappresentato con una serie di simboli
- Il segnale digitale è facilmente memorizzabile e processabile tramite le moderne unità di elaborazione, ha un range di variazione finito ed in qualche modo l'errore è definito a priori (errore di quantizzazione)
- Il tempo di campionamento (T) deve essere commisurato al contenuto frequenziale del segnale di interesse

▪ Conversione D/A

- Passaggio inverso alla conversione A/D
- Il risultato dell'elaborazione può essere convertito in analogico per pilotare gli attuatori

Sottosistemi

▪ Elaborazione digitale

- PC, laptop, Smartphone, scheda dedicata (microprocessore, microcontrollore, DSP,...)
- Filtraggio digitale
 - passa basso, passa banda, passa alto....
- Elaborazione del segnale
 - Estrazione parametri di interesse: ECG intervallo R-R, EMG inviluppo....
 - reazione alla misura nei sistemi ad anello chiuso
 - gestione dell'interfaccia di trasmissione (seriale, I2C, RS485)

▪ Trasmissione

- Wired (seriale, USB)
- Wireless (Bluetooth, ZigBee, ANT+)

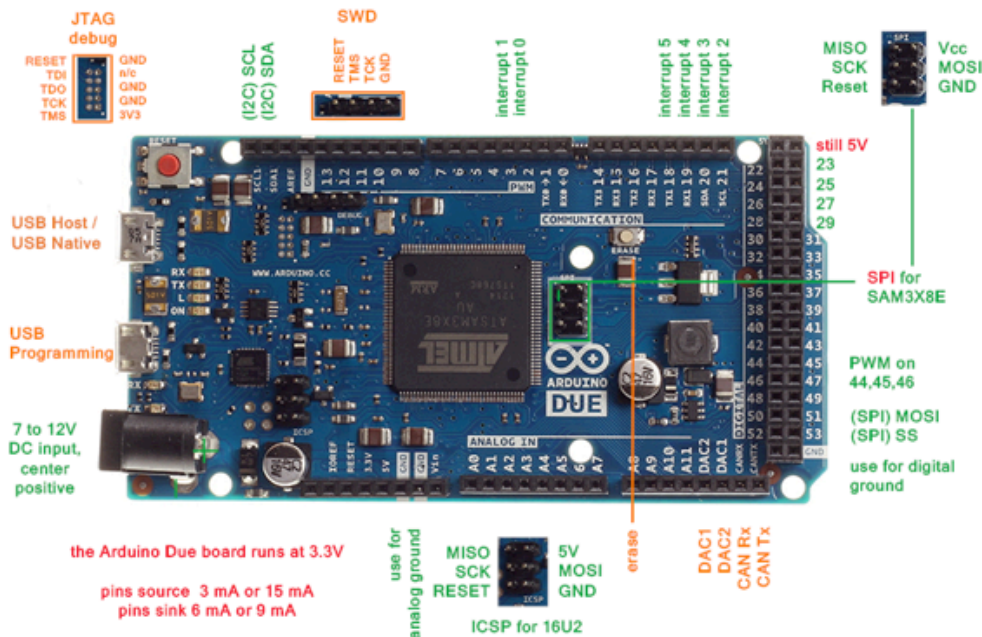
▪ Interfaccia utente

- display, graphical user interface (GUI), app...

Scheda Microprocessore ARDUINO

- Hardware, Software e prototipo di circuito

“Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing / Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.” (www.arduino.cc, 2006)

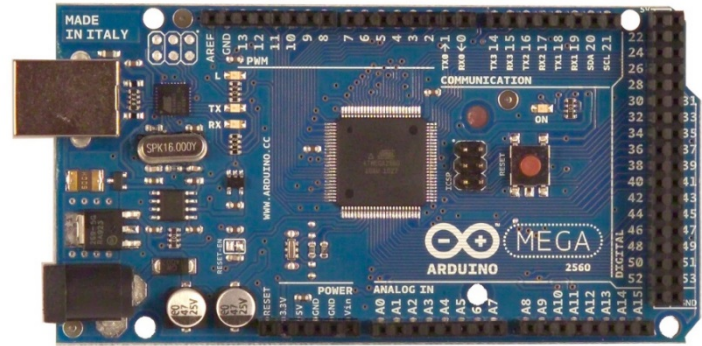
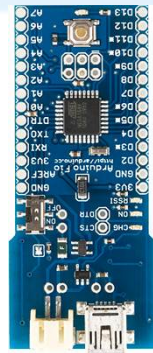
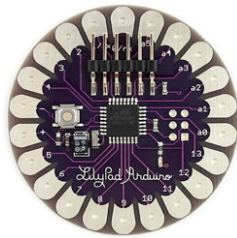


Com' è fatto Arduino

Principali Caratteristiche

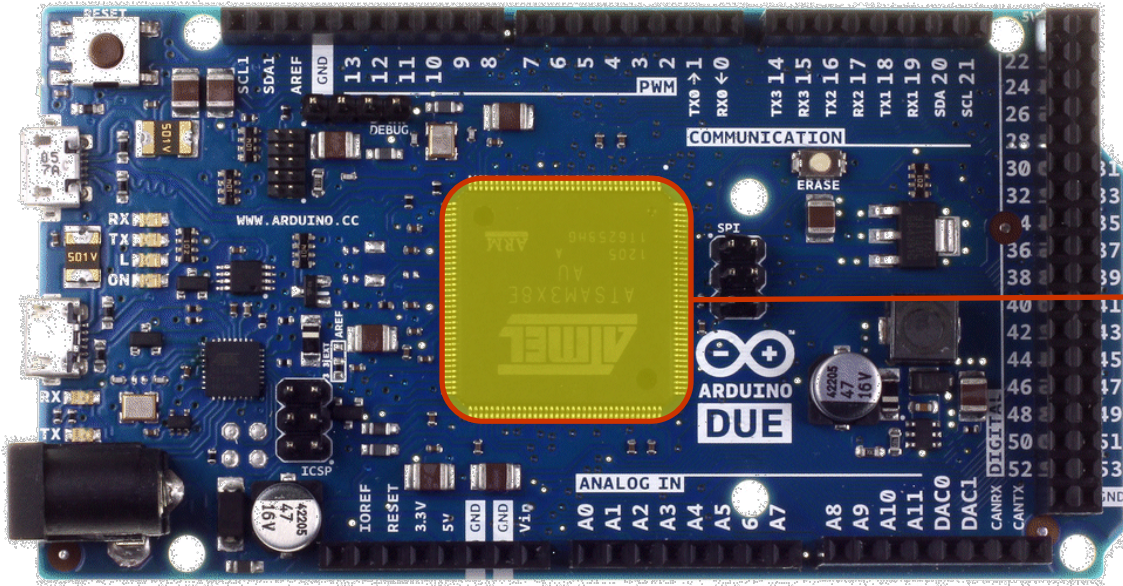
- Microcontrollore Atmel ARM Cortex-M3
 - 32 bit con frequenza di clock a 84 MHz
- Dispone di periferiche per la comunicazione seriale di tipo I2C e SPI, Ingressi analogici e digitali e uscite PWM
 - 54 pin digitali di I/O multifunzionali (16 uscite PWM)
 - 12 pin per ingressi analogici
 - 4UARTs (hardware serial ports)
- Memoria
 - Flash da 512KB
 - una EEPROM da 1KB
 - una memoria volatile SRAM da 96KB
- Alimentazione
 - fornita tramite cavo USB o da adattatore AC/DC fra 7 e 12 V da applicare tramite jack o tra i piedini VIN e GND del connettore POWER
 - Possibilità di prelevare 5V e 3,3 V da appositi piedini.

Perchè ARDUINO?



- Scheda elettronica che può essere programmata per ricevere input da diversi sensori e che può interagire con il mondo esterno attraverso il controllo di dispositivi attuatori.
- Supportato da Matlab e Simulink
- Sviluppo modulare: permette di estendere il progetto, partendo dal piccolo e crescendo pezzo per pezzo inserendo nuove funzionalità (Wireless
- Lo sviluppo è aiutato da una enorme community online con esempi e consigli.
- Economico

Com'è fatto ARDUINO



Arduino DUE è basato su un microcontrollore a 32 bit con core ARM (Atmel ARM Cortex-M3 SAM3X8E)

Il microcontrollore è il «cervello» di tutto il sistema, gestisce gli input, gli output, la comunicazione USB, la temporizzazione.

Necessita quindi di istruzioni → deve essere programmato!!!

Microcontrollore: Approfondimento

Il microprocessore è il nucleo centrale di un calcolatore; esso è l'unità di elaborazione dei dati e di controllo del funzionamento del calcolatore stesso e viene spesso indicato con la sigla CPU (Central Processing Unit). Posto da solo, il uP non è utilizzabile, infatti sia i dati che i programmi su cui il processore opera sono immagazzinati in un'unità di memoria esterna a causa della grande quantità di memoria richiesta.



Per applicazioni particolari, tipiche del controllo industriale si fa invece uso dei così detti Microcontrollori (uC).

Un microcontrollore è un sistema a microprocessore completo, integrato in un solo *chip*, progettato per ottenere la massima autosufficienza funzionale ed ottimizzare il rapporto prezzo-prestazioni per una specifica applicazione. I uC comprendono, oltre all'unità di calcolo, anche la memoria (RAM e ROM) e ulteriori periferiche di input/output (convertitori analogico/digitali, timer, interfaccia USB) a seconda dell'applicazione specifica.

Microcontrollore:

Approfondimento 2

I vantaggi dell' utilizzo dei micro:

- Sono richiesti meno dispositivi “discreti” per la realizzazione di un sistema
- Il sistema ha dimensioni ridotte
- Diminuiscono i costi
- Diminuisce il consumo di potenza
- Diminuisce la sensibilità all' ambiente (temperatura, EM, ...)
- Utilizza meno componenti, quindi più affidabile.
- Riconvertibilità del progetto (riprogrammabile)
- Protezione contro la copia
- Interfacciamento semplice con altri dispositivi (PC, LCD, ...)

Ma dove si usano???

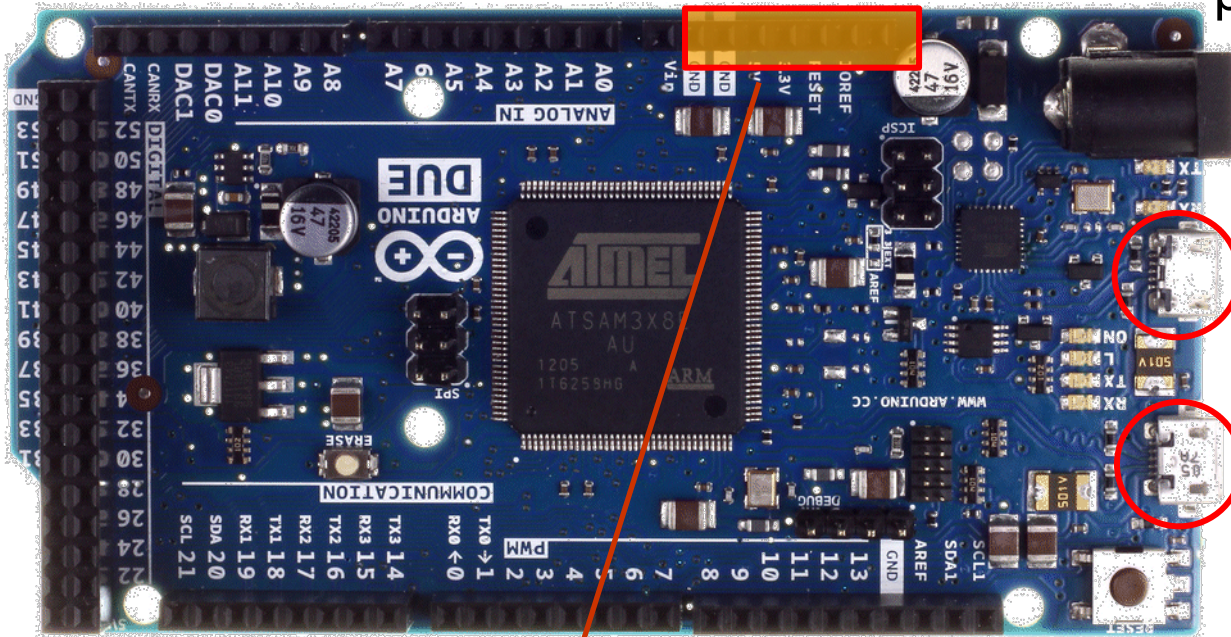
- Componenti PC: mouse, tastiere, modem, carica batterie...
- Orologi, calcolatrici
- Serrature per porte, sistemi d' allarme
- Automotive: in una BMW X5 sono contenuti più di 70 microcontrollori.

Arduino è basato su un microcontrollore, l' ATmega328

La grande community online e il fatto di essere OpenSource lo rendono molto più semplice ed intuitivo rispetto al normale utilizzo di microcontrollori (PIC, ARM...)

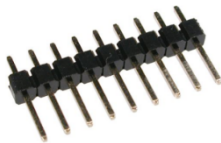
Com'è fatto ARDUINO

Alimentazione e
programmazione da USB



Sezione POWER:
Alimentazione da V-in (7 - 20V)

Alimentazione da Jack (7 - 20V)



Pin Analogici

- Arduino Due possiede 12 ingressi analogici (Pin da A0 a A11) direttamente connessi al convertitore Analogico-Digitale (ADC)
 - ogni pin presenta una risoluzione a 12 bit (4096 livelli)
 - per mantenere la compatibilità con le altre schede Arduino la risoluzione è fissata a 10 bit (è possibile variarla)
 - su questi pin è ammessa una tensione massima pari a 3,3V



Pin Digitali

- Ognuno dei 54 pin Digitali I/O dell'Arduino può essere utilizzato sia come input che come output
 - operano ad una tensione di 3.3V e possono fornire fino a 15mA di corrente
- Alcuni di questi pin hanno funzioni specifiche:
- Serial: pin 0 (RX) e pin 1 (TX). Sono rispettivamente il pin di trasmissione e ricezione per la comunicazione seriale. Possono essere utilizzati per connettere un modulo di trasmissione wireless (Bluetooth, WiFi, etc..)
- DAC1 and DAC2: questi pin forniscono uscite analogiche con risoluzione a 12 bit (4096 livelli)
 - Possono essere usati per pilotare un attuatore o per fornire un valore di tensione al front-end analogico
 - Range di valori standard: da 0.55 V fino a 2.75 V

ARDUINO e MATLAB

- Con MATLAB® Support Package for Arduino® Hardware, è possibile usare MATLAB per interagire direttamente con Arduino. Il pacchetto software permette di eseguire funzioni tipo:
 - acquisire segnali analogici e digitali dei sensori collegati a Arduino
Acquire analog and digital sensor data from your Arduino board
 - controllare attuatori attraverso le porte digitali e le uscite PWM
 - Comunicare con una scheda Arduino via USB o soluzioni wireless (tipo WiFi)

ARDUINO e SIMULINK

- Con [Simulink® Support Package for Arduino® Hardware](#), è possibile usare Simulink per sviluppare e simulare algoritmi che lavorano direttamente su Arduino:
- Il support hardware permette:
- Sfruttare le potenzialità di Simulink di lavorare a blocchi per la configurazione delle diverse funzionalità di Arduino
 - elaborare e registrare dati dei sensori e attuatori
 - Usare i blocchi UDP and TCP/IP per far comunicare smartphone con Arduino (usando shield Wi-Fi e Ethernet) External mode for interactive parameter tuning and signal monitoring as your algorithm runs on the device (not available on some boards – see list below)