

**ACHILLE  
CANNAVALE**

# **APPUNTI ARCHITETTURE DEI CALCOLATORI**

**2022**

**CIAO! QUESTI APPUNTI SONO  
FRUTTO DEL MIO STUDIO E  
DELLA MIA INTERPRETAZIONE,  
QUINDI POTREBBERO  
CONTENERE ERRORI, SVISTE O  
COSE MIGLIORABILI. BUONO  
STUDIO 📖 ✎**

# Architetture dei Calcolatori

Riassunto da

**Achille Cannavale**



# Indice

<b>1</b>	<b>Sistemi Embedded</b>	<b>5</b>
1	I Sistemi Embedded . . . . .	5
1.1	Micro Control Unit . . . . .	5
1.2	System on Chip . . . . .	5
1.3	Microprocessori . . . . .	5
2	Protocolli di Comunicazione . . . . .	6
2.1	I2C (Inter Integrated Circuit) . . . . .	6
2.2	SPI (Serial Peripheral Interface) . . . . .	6
2.3	UART (Universal Asynchronus Protocol) . . . . .	7
3	Gestione degli Eventi . . . . .	8
3.1	Polling . . . . .	8
3.2	Interrupt . . . . .	8
4	Segnale PWM (Pulse Width Modulation) . . . . .	10
5	Timer . . . . .	10
6	Il RTOS (Real Time Operating System) . . . . .	11
6.1	REAL TIME . . . . .	11
6.2	SISTEMA OPERATIVO . . . . .	11
7	FreeRTOS . . . . .	13
<b>2</b>	<b>Sensoristica</b>	<b>15</b>
1	Che cosa è un Sensore? . . . . .	15
2	Campi di un Sensore . . . . .	16
3	Incertezze . . . . .	17
3.1	Errore di Non Linearità . . . . .	17
3.2	Sensibilità . . . . .	17
3.3	Ripetibilità . . . . .	17
3.4	Stabilità . . . . .	17
3.5	Isteresi . . . . .	18
4	Trasduttore per Sensori Resistivi . . . . .	19
4.1	Generatore di Corrente . . . . .	19
4.2	Partitore di Tensione . . . . .	20
4.3	Potenziometro in configurazione Push-Pull . . . . .	21

	4.4	Ponte di Weathstone . . . . .	21
5		Convertitore AD . . . . .	24
	5.1	ADC nel dettaglio . . . . .	24
	5.2	Parametri Caratteristici di un ADC . . . . .	25
	5.3	Decimazione . . . . .	25
	5.4	Rapporto Segnale Rumore . . . . .	25
	5.5	Media Bit Scartati . . . . .	26
	5.6	Cambiare Frequenza . . . . .	26
6		Acquisizione Multiplexata . . . . .	27
	6.1	Interchannel Delay . . . . .	28
7		A Massa Vs Floating . . . . .	29
	7.1	A Massa . . . . .	29
	7.2	Differenziali (o Floating) . . . . .	29

# Capitolo 1

## Sistemi Embedded

### 1 I Sistemi Embedded

Un sistema **embedded** è dedicato a particolari funzioni, in cui può non essere presente un sistema operativo.

Un sistema embedded può essere basato su un **microprocessore** o un **microcontrollore**.

La differenza sta nel fatto che il microprocessore ha per l'appunto solo un microprocessore mentre, il microcontrollore ha: la **CPU**, una **memoria flash**, la **RAM**, **periferiche** etc.

I sistemi embedded funzionano grazie a:

- **MCU** (Micro Control Unit)
- **SoC** (System on Chip)
- **CPU** (Microprocessori)

#### 1.1 Micro Control Unit

Il **Micro Control Unit** è un dispositivo integrato che rileva eventi e comunica con il mondo esterno.

#### 1.2 System on Chip

Il **System on Chip** è circuito integrato con vari componenti elettronici interconnessi, che viene affiancato da altre periferiche.

#### 1.3 Microprocessori

La **CPU** elabora dati ed esegue istruzioni attraverso operazioni aritmetiche.

## 2 Protocolli di Comunicazione

### 2.1 I2C (Inter Integrated Circuit)

Il Protocollo I2C è un protocollo **Master-Slave Half-Duplex**, caratterizzato da due linee di comunicazione:

- **SDA** che invierà i dati
- **SCL** invece fungerà da orologio dato che è una comunicazione sincrona.

Per avviare una comunicazione, il Master farà un **pull down** sul canale **SDA** mentre il canale **SCL** è **alto**, questo segnerà lo **START**.

Fatto ciò il Master invierà sul canale **SDA** i dati, corredati dal **bit ACK** che serve a confermare la corretta ricezione, ma solo lo Slave interessato li riceverà grazie allo **Slave-Address** identificativo inviato in precedenza.

Gli Slave possono rispondere al Master, tuttavia **NON** possono iniziare una comunicazione.

Infine il Master farà un **pull up** mentre il canale **SCL** è **alto** per terminare la comunicazione.

### 2.2 SPI (Serial Peripheral Interface)

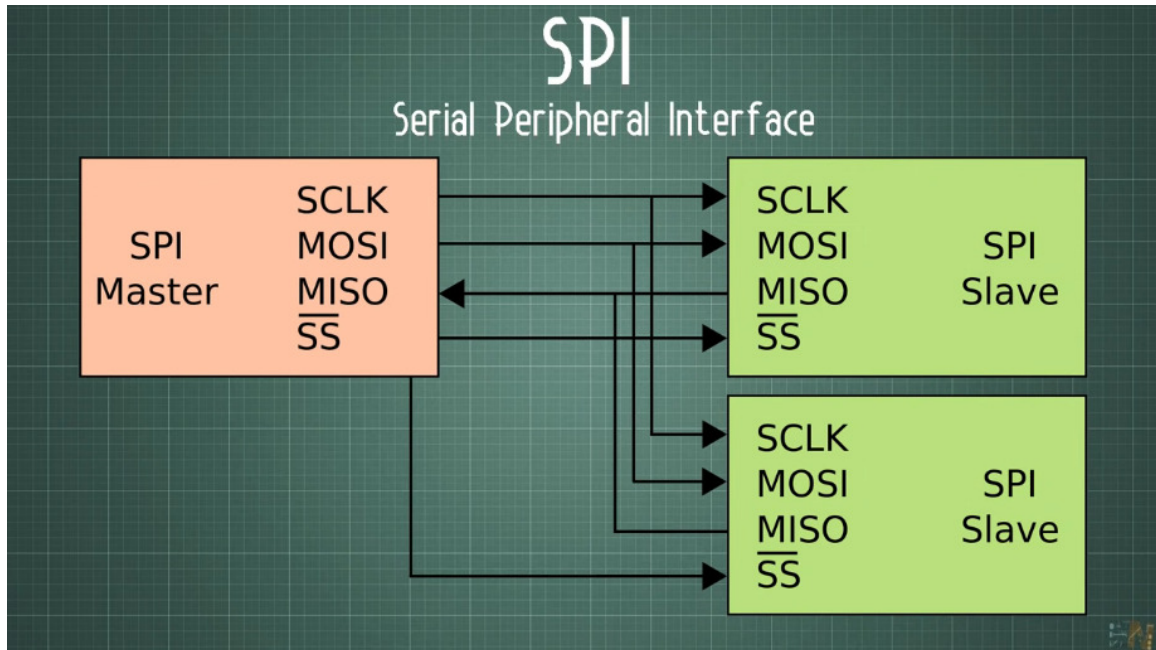
Il Protocollo **SPI** è un protocollo **Master-Slave Full-Duplex** che oltre ad avere il collegamento relativo al Master (**SCK**) ha tre collegamenti in più:

- **MOSI** (Master Output Slave Input)
- **MISO** (Master Input Slave Output)
- **CS** (Cavo di Selezione del Chip)

Si possono avere più Slave e un solo Master.

Qui non usiamo lo **Slave-Address**, a differenza utilizzeremo proprio il **cavo di Selezione del Chip**.

Per iniziare una trasmissione il Master fa un **pull down** sul **Chip Select**, facendo partire il **Clock** e i dati sul canale **MOSI**.



## 2.3 UART (Universal Asynchronous Protocol)

Il Protocollo di comunicazione **UART** utilizza due soli fili, per creare una comunicazione **asincrona mono e bidirezionale**.

Ma come capire quando il Trasmittente inizia e finisce di comunicare?

Supponendo che Tx e Rx lavorino con le stesse configurazioni:

- **Transmission speed**
- **Data length**
- **START and STOP bits**

Solitamente Tx trasmette sempre un segnale alto, quindi lo **START** sarà rappresentato da un **bit down** (detto **Pull Down Bit**). Successivamente grazie alla **Transmission speed** e alla **Data Length**, capirà quando dovrà smettere di leggere, ovvero quando ha letto Data Length bits.

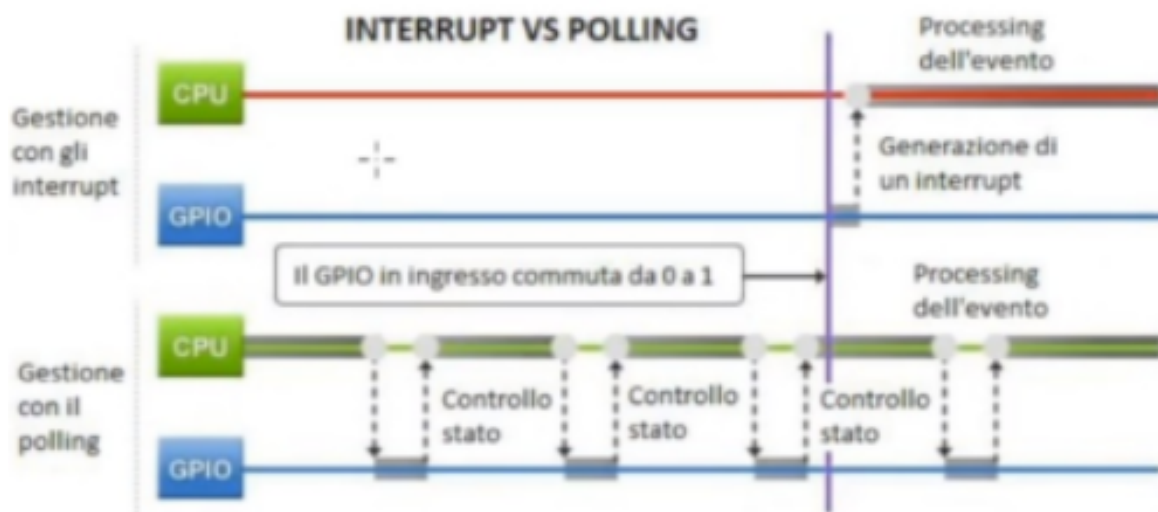
Per esempio Arduino usa uno di questi **GPIO pin** per comunicare con **Arduino IDE**.



### 3 Gestione degli Eventi

La miriade di periferiche che possiede il modulo ESP32 vengono gestite attraverso gli **eventi** con due tecniche:

- Polling
- Interrupt



#### 3.1 Polling

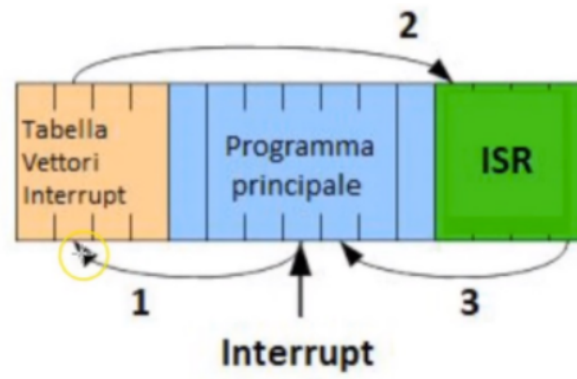
La tecnica **Polling** serve per gestire gli eventi delle periferiche, monitorando l'evento periodicamente, **non garantendo così la latenza**.

#### 3.2 Interrupt

La tecnica **Interrupt** serve per gestire gli eventi delle periferiche, cercando di **ridurre al minimo la latenza tra evento e gestione dello stesso**.

L'**Interrupt** interrompe il flusso del programma in modo **asincrono**.

Il processore cerca nella **Vector Table Entry** cedendo poi il controllo all'**ISR** associato che gestirà l'interrupt. Il flusso di controllo riprenderà da dove si è interrotto in precedenza.



Gli **Interrupt** hanno una priorità associata all'importanza dell'evento associato.

## 4 Segnale PWM (Pulse Width Modulation)

Il **segnale PWM** non è altro che una serie di **onde quadre**, utilizzato per variare la potenza elettrica erogata, agendo sulla durata dell'impulso.

Normalmente gli impulsi vengono generati ad una frequenza fissa, variando il **Duty Cycle**.

Il **Duty Cycle** è:

$$\frac{\textit{Durata dell'impulso nello stato alto}}{\textit{Durata totale dell'impulso}}$$

Questo metodo di modulazione è usato principalmente per il controllo dei **led ad incandescenza**.

## 5 Timer

Il timer hardware è presente in ogni microcontrollore e la sua funzionalità principale è quella di misurare lo scorrere del tempo conteggiando gli impulsi associati al clock della CPU.

## 6 Il RTOS (Real Time Operating System)

Il termine **RTOS** deriva dall'unione **Real Time** e **Operating System**.

### 6.1 REAL TIME

Indice che il sistema **Embedded** deve reagire agli eventi con tempi di risposta minimi. (Garantire una latenza massima della gestione degli **Interrupt**).

### 6.2 SISTEMA OPERATIVO

Si intende l'insieme di programmi che fungono da interfaccia tra **Hardware** e i **Programmi Applicativi**.

I **RTOS** si dividono in tre categorie:

- **Hard Real Time**
- **Firm Real Time**
- **Soft Real Time**

Un **RTOS** è formato principalmente da:

- **Kernel**
- **Interfaccia di Rete**
- **Protocolli di Comunicazione**
- **Driver per Periferiche**

Il **Kernel**

Il **Kernel** è un componente fondamentale che fornisce questi principali servizi:

- **Gestione dei Task**
- **Gestione degli Interrupt e degli Eventi**
- **Sincronizzazione dei Task e IPC**
- **Gestione dei dispositivi I/O**

- Gestione della **memoria**
- Gestione dei **Timer**

## 7 FreeRTOS

**FreeRTOS** è un **RTOS** che ha il compito di avere un **Kernel** per **Sistemi Embedded** di dimensioni compatte.

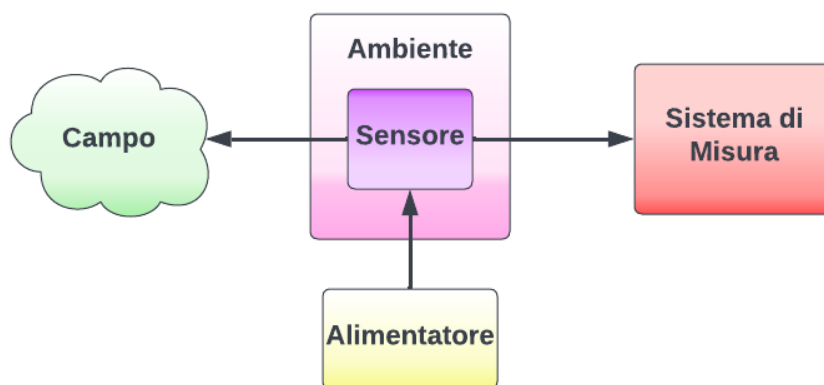
Uno dei compiti del **FreeRTOS** è la gestione dei **Task**, degli **Interrupt** e della **memoria**.



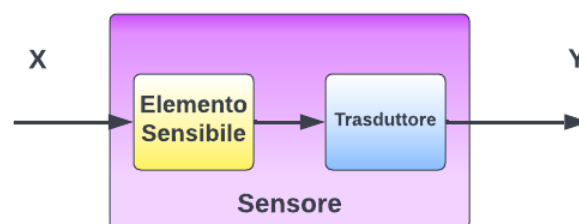
## Capitolo 2

# Sensoristica

### 1 Che cosa è un Sensore?



Un sensore è un dispositivo che trasforma un'informazione presa dal Campo in una differenza di potenziale.



Quindi abbiamo che:

$$y = f(x)$$

Ma  $y$  per noi non ha alcun interesse, dato che siamo interessati alla misurazione  $x$ , quindi più correttamente siamo interessati a:

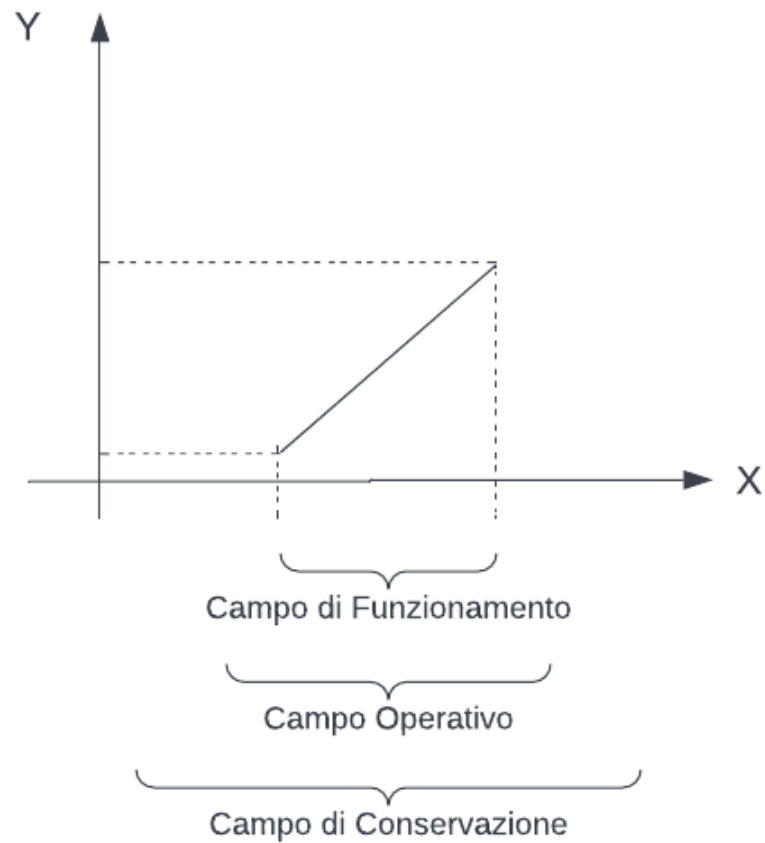
$$x = f^{-1}(y) = g(y)$$



## 2 Campi di un Sensore

Ora analizzeremo i Campi di Utilizzo dei Sensori:

- Campo di Funzionamento
- Campo Operativo
- Campo di Conservazione



### 3 Incertezze

Purtroppo non esiste una corrispondenza lineare tra  $y$  e  $x$ , dato che vi sono molte incertezze, tra le quali:

- Incertezza sul misurando  $u_x$  a causa della variabilità delle grandezze.
- Incertezza Ambientale  $u_a$  a causa del fatto che il sensore sia immerso in un ambiente.
- Incertezza Power Supply  $u_{ps}$  dovuta all'alimentazione.
- Incertezza sul Sistema di Misura  $u_m$  a causa dei fenomeni parassiti.

#### 3.1 Errore di Non Linearità

Data la difficoltà di esprimere una funzione che lega  $Y$  a  $X$ , i costruttori di sensori danno nel datasheet una funzione lineare del genere:

$$y = \alpha x + q$$

ma facendo ciò si compie un errore, che viene detto Errore di Non Linearità:  $u_{NLE} = \frac{E_{NLE}}{\sqrt{3}}$

#### 3.2 Sensibilità

La Sensibilità indica quanto varia un'uscita rispetto a quanto varia l'ingresso:  $S = \frac{dy}{dx}$

Il grafico ideale della sensibilità sarebbe una retta ( $S = 0$  sempre), tuttavia il grafico reale è una curva con punti di buona sensibilità e punti di bassa sensibilità.

Quindi questo ci crea un'incertezza  $u_s = \frac{\Delta S}{\sqrt{3}}$

#### 3.3 Ripetibilità

Fissata una misurazione  $x$ , valuto la grandezza  $y$  nel tempo. Le variazioni che nel tempo la grandezza  $y$  subisce, rappresentano una nuova incertezza, ovvero la Ripetibilità:  $u_r$ .

#### 3.4 Stabilità

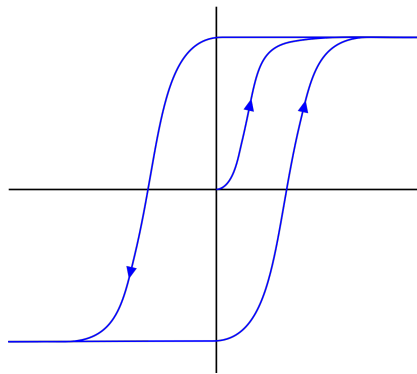
La Stabilità è un valore importante per misurazioni di lungo periodo ed indica l'invecchiamento del sistema.

Quindi anche questo indicatore rappresenta un'incertezza:  $u_{st}$ .

### 3.5 Isteresi

L'Isteresi è un'informazione che ci dice che la curva reale del sensore non dipende solo dal punto in considerazione, ma anche dal modo in cui ci si arriva:

- da una misurazione dal basso
- da una misurazione dall'alto



E ciò rappresenta un'incertezza:  $u_{his}$

Quindi unendo tutte queste incertezze più molte altre avremo:

$$u_{sistema} = \sqrt{u_{NLE}^2 + u_s^2 + u_r^2 + u_{his}^2 + u_{st}^2 + \dots + \underbrace{u_{\sigma_f}^2}_{\text{Incertezza di Frequenza}}}$$

## 4 Trasduttore per Sensori Resistivi

Possiamo categorizzare i Sensori in diversi modi, ma il più usato è caratterizzarli per la grandezza che fanno variare:

- Sensori Resistivi
- Sensori Capacitivi
- Sensori Induttivi

Analizzeremo in questa sezione i Sensori Resistivi<sup>1</sup> e i loro Trasduttori.

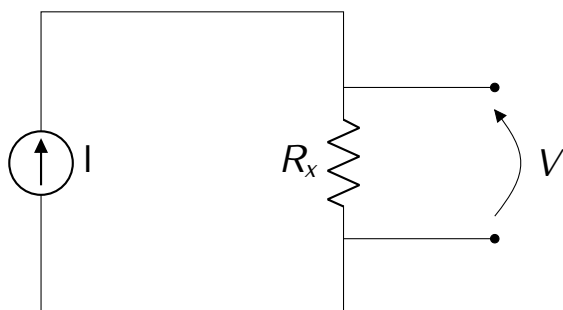
Data  $\Delta x \Rightarrow \Delta R \Rightarrow \Delta V$  che posso portare in ingresso ad un qualunque Sistema di Misura.

Ma come faccio a passare da una variazione di Resistenza ad una variazione di Tensione????

### 4.1 Generatore di Corrente

Potremmo alimentare  $R_x$  attraverso un Generatore di Corrente Costante:

$$I = \text{cost} \Rightarrow \Delta V = \Delta R \cdot I$$



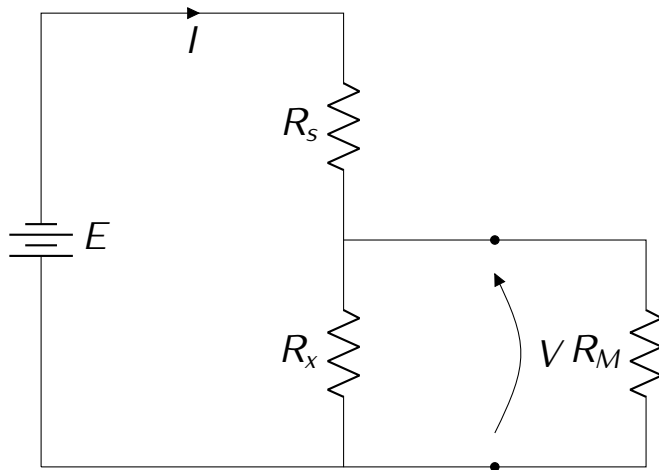
dove  $V = R_x I$ .

Tuttavia è una soluzione poco pratica dato che Generatore di Corrente ha un costo eccessivo.

---

<sup>1</sup>Ricordiamo che  $R = \rho_s^l (1 + \alpha \Delta T)$

## 4.2 Partitore di Tensione



Dove  $V_x = \frac{R_x // R_M}{R_x // R_M + R_s} \cdot E$

Tuttavia NON è Lineare perchè  $R_x$  sta sia sopra che sotto e non è neanche Sensibile, ma è sempre meglio del Generatore di Corrente che ha un costo eccessivo.

### Sensibilità

Per eliminare la NON Sensibilità possiamo supporre:

$$R_x \ll R_m \implies R_x \text{ dipende dal tipo di Sistema di Misura Usato } (R_M)$$

Se vale ciò allora:

$$V_x = \frac{R_x}{R_x + R_s} \cdot E$$

### Linearità

Per eliminare la NON Linearità supponiamo:

$$R_x = R_0 + \Delta R_x \implies \begin{cases} V_0 = \frac{R_0}{R_0 + R_s} E \\ V_x = \frac{R_0 + \Delta R_x}{(R_0 + \Delta R_x) + R_s} E \end{cases}$$

Ciò implica che:

$$V_x = \left[ \frac{R_0}{(R_0 + \Delta R_x) + R_s} E + \frac{\Delta R_x}{(R_0 + \Delta R_x) + R_s} E \right]$$

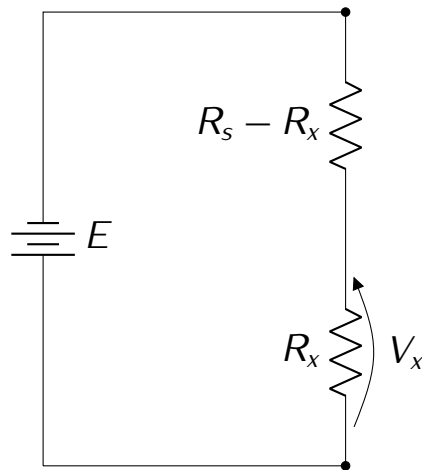
Supponendo:

- $\Delta R_x \ll R_0$
- $R_s \gg R_0$

$$V_x \approx V_0 + \frac{\Delta R_x}{R_0 + R_s} E = \alpha \Delta R_x$$

Quindi il Partitore di Tensione di per sè NON è Lineare, ma se considero una piccola variazione ( $\Delta R_x$ ) restituisce un risultato Lineare!

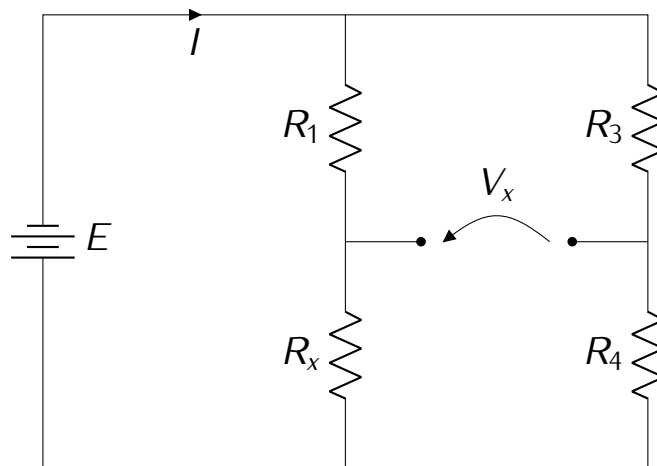
### 4.3 Potenzimetro in configurazione Push-Pull



$$\text{Con } V_x = \frac{R_x}{R_x + (R_s - R_x)} E = \frac{R_x}{R_s} E = \alpha R_x$$

Questo metodo rimane Lineare anche per grandi variazioni!!

### 4.4 Ponte di Weathstone



$$V_x = \left[ \frac{R_x}{R_1 + R_x} - \frac{R_4}{R_3 + R_4} \right] E$$

Ponendo:

- $R_x = R_0 + \Delta R$
- $R_1 = R_3 = R_4 = R_0$

$$V_x = \left[ \frac{R_0 + \Delta R}{R_0 + R_0 + \Delta R} - \frac{R_0}{2R_0} \right] E$$

Se consideriamo  $\Delta R \ll R_0$  e  $---$  allora:

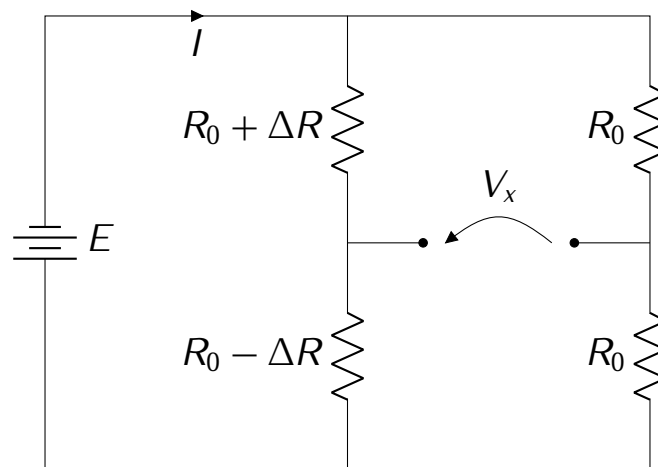
$$V_x \approx \left[ \frac{R_0}{2R_0} + \frac{\Delta R}{2R_0} - \frac{1}{2} \right] E = \frac{\Delta R}{2R_0} E$$

La qualità di questo metodo è che a differenza del Potenziometro in cui dovevo fare prima una misurazione per  $R_0$  (a regime) e poi la misurazione effettiva  $R_x$ , qui invece basta fare una sola misura.

Tuttavia funziona per piccole variazioni.

### Linearità

Per ottenere la Linearità utilizziamo uno schema Push-Pull

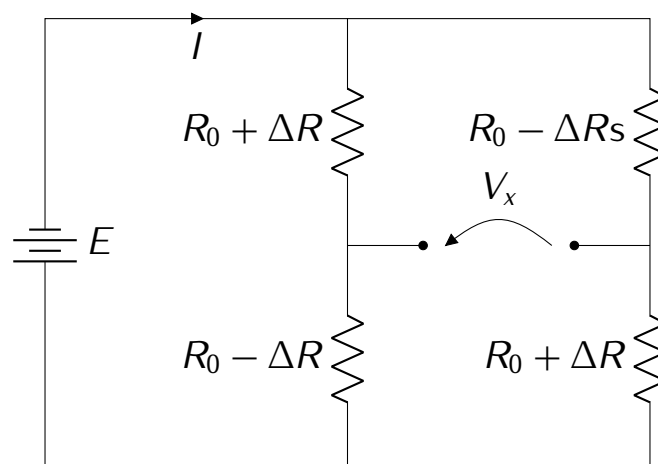


$$V_x = \left[ \frac{R_0 - \Delta R}{2R_0} - \frac{1}{2} \right] E = -\frac{\Delta R}{2R_0} E$$

Facendo così, il Ponte di Weathstone funzionerà anche per grandi variazioni.

### Sensibilità

Per ottenere invece la Sensibilità dovremmo utilizzare la seguente conformazione:



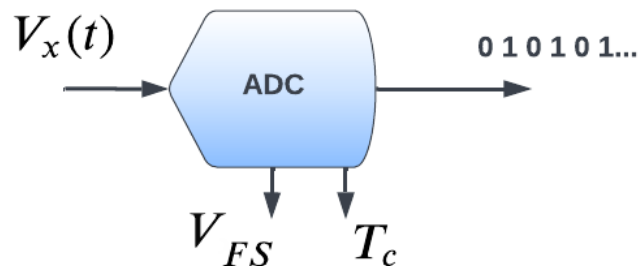
$$V_x = \left[ -\frac{\Delta R_x}{R_0} \right] E$$

Quindi abbiamo una sensibilità DOPPIA rispetto a prima!



## 5 Convertitore AD

Esaminiamo ora le varie caratteristiche che rappresentano un **Convertitore Analogico Digitale**.



Esso prende in ingresso una Tensione  $V_x$ :

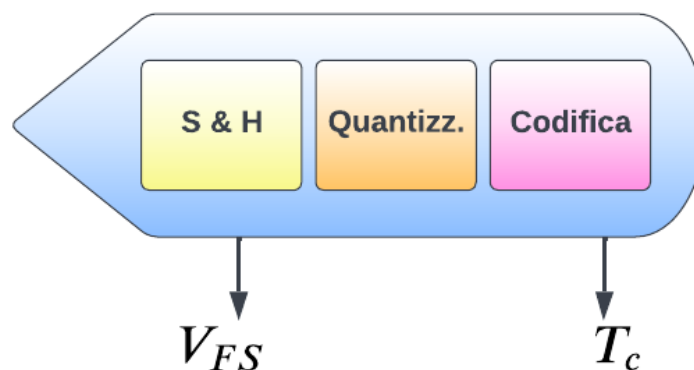
- Continua nel tempo
- Continua nelle ampiezze

A cui associa in uscita una **stringa di bit**:

- Discreta nel tempo
- Discreta nelle ampiezze

### 5.1 ADC nel dettaglio

Se entriamo all'interno di un **Convertitore Analogico Digitale** possiamo distinguere tre "azioni" fondamentali che vengono compiute:



- Campionamento Sample And Hold
- Quantizzazione
- Codifica

## 5.2 Parametri Caratteristici di un ADC

Esistono alcuni **parametri** a cui dobbiamo subito far caso quando valutiamo un Convertitore AD e sono:

$$\left\{ \begin{array}{l} V_{FS} \\ N \end{array} \right\} \Delta = \frac{V_{FS}}{2^N} < \text{Risoluzione Desiderata}$$

$$\left\{ \begin{array}{l} T_c \end{array} \right\} \Rightarrow f_s = \frac{1}{T_c}$$

Dove:

- $\Delta$  è la distanza tra un livello e l'altro
- $N$  è il numero di bit

## 5.3 Decimazione

In un Microcontrollore, nel caso in cui si volesse operare ad un tempo di campionamento diverso da  $T_c$  si può ricorrere alla **Decimazione**, ovvero scartare dei campioni, campionando a **multipli** di  $T_c$ .

## 5.4 Rapporto Segnale Rumore

Supponiamo di avere una tensione che varia in ampiezza tra  $V_{FS}/2$  e  $V_{FS}/2$ .

Scriviamo quindi così il **Valore Efficace**:

$$V_{EFF} = \frac{V_{FS}}{2} \cdot \frac{1}{\sqrt{2}}$$

Ora assumendo un errore di sovraccarico prossimo allo zero, possiamo scrivere il rumore a cui il segnale è assoggettato così:

$$\sigma_x = \frac{\Delta}{\sqrt{12}} = \frac{V_{FS}}{2^N} \cdot \frac{1}{\sqrt{12}}$$

Quindi utilizzando la formula dell' $SNR = 20 \log_{10}(\frac{V_{EFF}}{Noise})$ , potremmo mettere in relazione l' $SNR$  ed  $N$  in questo modo:

$$\begin{aligned} \Rightarrow SNR &= 20 \log_{10} \left( \frac{\cancel{V_{FS}}}{2} \cdot \frac{1}{\sqrt{2}} \cdot \frac{2^N \sqrt{12}}{\cancel{V_{FS}}} \right) = \\ &= 20 \log_{10}(2^N) + 20 \log_{10} \left( \frac{\sqrt{12}}{2\sqrt{2}} \right) = \\ &\approx 6.02 \cdot N + 1.71 \end{aligned}$$

## 5.5 Media Bit Scartati

Mettiamo il caso di voler prendere un campione ogni 9.

Gli altri 9 però, non conviene "buttarli", ma **mediarli**:

$$X = \frac{\sum_{i=1}^1 0x_i}{10}$$

## 5.6 Cambiare Frequenza

Nel caso in cui volessimo utilizzare una frequenza diversa da quella di campionamento, avremo un aumento di Bit disponibili secondo la seguente legge:

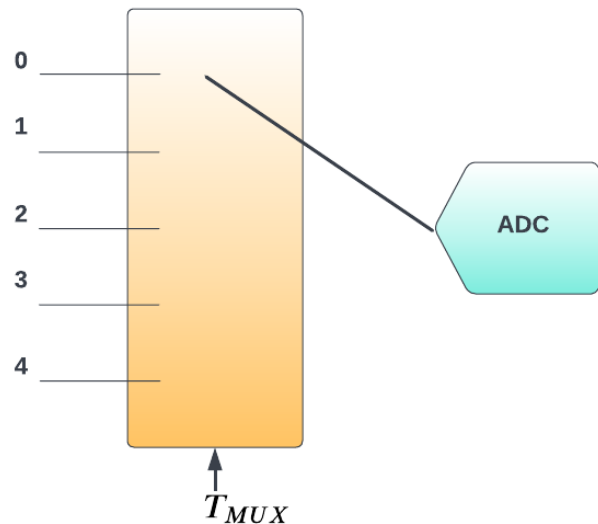
$$N_{new} = N_{old} + \frac{1}{2} \log_2(M)$$

dove:

$$M = \frac{f_c}{f_{scelta}}$$

## 6 Acquisizione Multiplexata

In questa configurazione ogni  $T_{MUX}$  secondi il **Multiplexer** commuta su tutti i canali.



Quindi alla fine avrò una matrice di questo tipo:



<b>Canale 0</b>	....	....	....
<b>Canale 1</b>	....	....	....
<b>Canale 2</b>	....	....	....
<b>Canale 3</b>	....	....	....
<b>Canale 4</b>	....	....	....

Ma evidentemente sto compiendo degli errori di approssimazione dato che per esempio sto dicendo che il primo sample del canale 4 sia a tempo  $T = 0$ , anche se in realtà è avvenuto a tempo  $T = 4$ .

In poche parole, **sto perdendo la relazione di fase dei canali.**

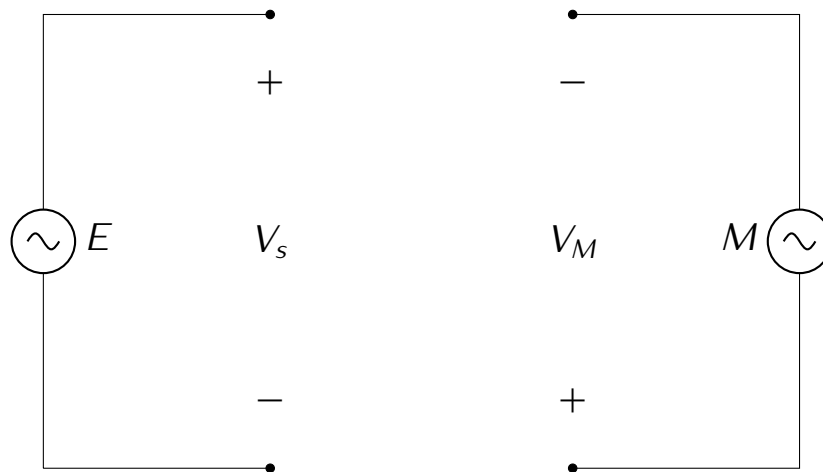
Dobbiamo quindi aggiungere un **nuovo parametro caratterizzante** dell'ADC:

## 6.1 Interchannel Delay

L'Interchannel Delay non è altro che il **ritardo di fase** tra due Canali Adiacenti:

$$\underbrace{\theta}_{\text{Fase}} = \frac{2\pi(\Delta T + \text{Interchannel Delay})}{T}$$

## 7 A Massa Vs Floating



A sinistra abbiamo il circuito di un **Sensore**, mentre a destra il circuito di uno **Strumento di Misura**.

Ora effettueremo un ragionamento che vale per entrambi i componenti.

### 7.1 A Massa

Un Sensore o uno Strumento di Misura, possono essere "A Massa", ovvero con un terminale fissato ad una certa tensione, mentre l'altro variabile.

Così facendo, per valutare la Differenza di potenziale occorrerà effettuare una sola misurazione, quella sul terminale variabile.

### 7.2 Differenziali (o Floating)

Un problema che la costituzione A Massa presenta è quello del **Rumore di Modo Comune**.

Esso è un problema che è sempre presente e che verrà messo in conto nella misurazione della tensione sul terminale libero.

Questo non rappresenta un problema però nella conformazione **Differenziale**, nella quale entrambi i terminali sono variabili!

Una volta misurata la tensione su ambo i capi (corredata dal rumore), calcoleremo la differenza di potenziale così:

$$(V_A + \mathcal{N}) - (V_B + \mathcal{N})$$

Tuttavia anche questo schema presenta un problema, ovvero:

**Bisogna chiedersi sempre quale sia la Massima Tensione che il  
Microcontrollore può utilizzare!!**

**Esempio**

Supponendo di volere una differenza di potenziale di 1 volt, con un sistema Differenziale potremmo avere:

$$V_A = 1.5V \quad V_B = 0.5V$$

Tuttavia dato che entrambi i capi sono liberi potremmo anche avere:

$$V_A = 201V \quad V_B = 200V$$

Dato che la differenza fa sempre 1 V, ma nel secondo caso molto probabilmente il microcontrollore si romperebbe...