

Istituto Professionale per L'industria L'Artigianato
"Antonio Guastaferro"

63074 SAN BENEDETTO DEL TRONTO (AP)

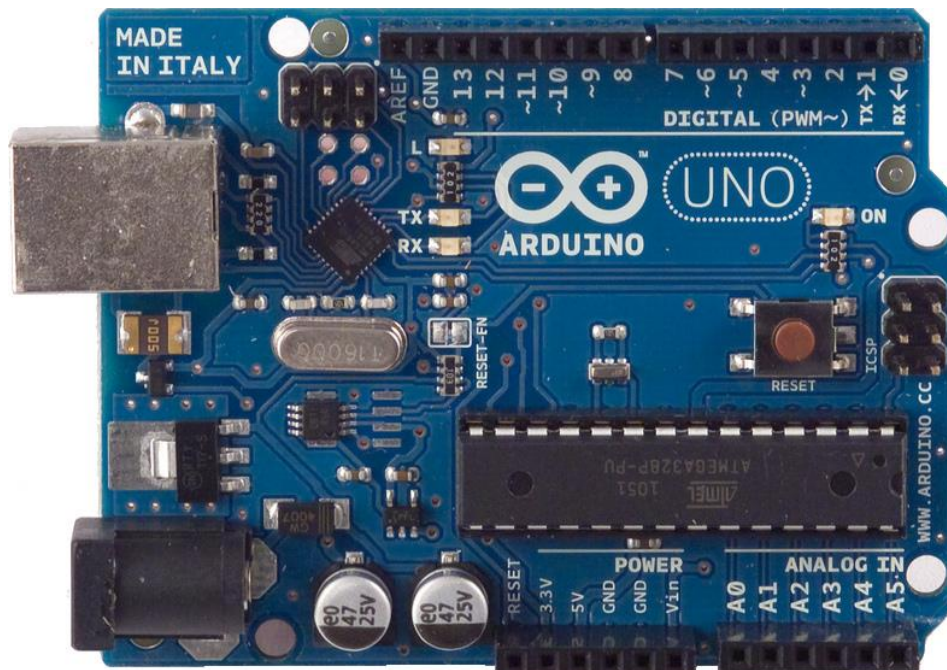
Classe : 4A_IPAI 5A_IPAI
A.S. : 2021-2022
Docente : Tufoni Franco
Disciplina : Tecnologie elettriche-elettroniche, dell'automazione e applicazioni

Arduino

Convertitore A/D

Hardware - Software

Applicazioni



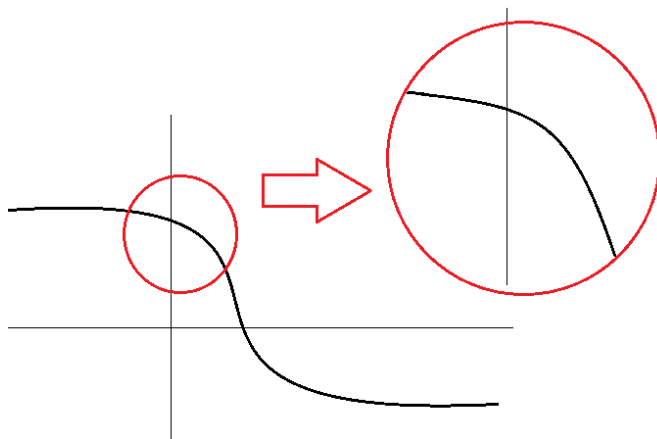
Vers. 3.1

Open Source

Conversione Analogico-Digitale (Fonte: <http://www.elemania.altervista.org/>)

Grandezze analogiche

Una grandezza viene detta analogica, quando essa può assumere con continuità qualsiasi valore (all'interno di un certo intervallo) in qualsiasi istante di tempo.

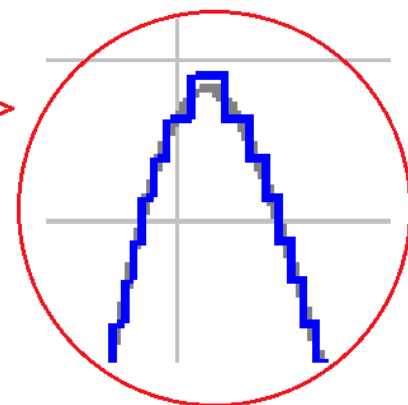
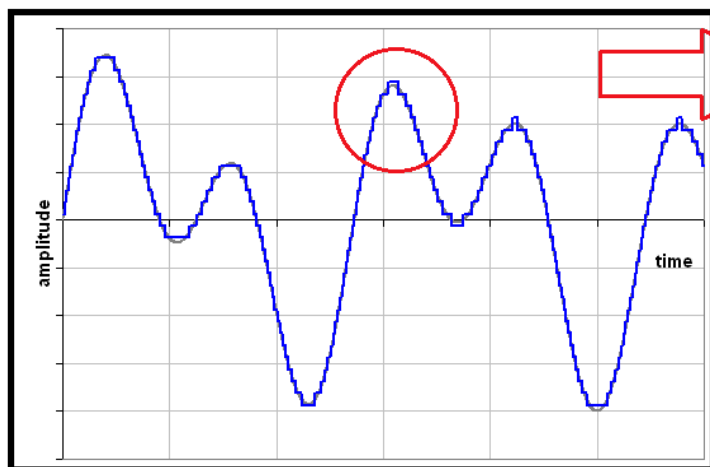


Per esempio la temperatura è una grandezza analogica, in quanto qualsiasi valore di temperatura (sopra lo zero assoluto) è possibile. In altre parole: dati due valori di temperatura, comunque vicini, esiste sempre un terzo valore compreso fra i due. Per esempio fra 27 °C e 28 °C c'è 27,5 °C; fra 27,5 °C e 27,6 °C c'è 27,55 °C e così via all'infinito. In generale una grandezza analogica può

essere suddivisa arbitrariamente quante volte si vuole. Inoltre le grandezze analogiche, come la temperatura, variano nel tempo con continuità, nel senso che non esistono "istanti di variazione" o "di commutazione" distinti.

Grandezze quantizzate e quanto

Una grandezza viene detta quantizzata quando due suoi valori consecutivi sono separati da un intervallo che non contiene altri valori validi. Per esempio, il prezzo di una merce espresso in euro è una grandezza quantizzata, in quanto i suoi valori non possono variare per meno di un centesimo di euro.



L'intervallo minimo fra due valori consecutivi di una grandezza digitale, viene detto quanto. Per esempio nel caso dei prezzi in euro, il quanto è pari a un centesimo di euro.

Il quanto rappresenta la più piccola variazione possibile della grandezza digitale (che viene pertanto detta anche quantizzata).

Grandezze discrete

Una grandezza si dice discreta (o discretizzata) quando assume valori solo in corrispondenza di certi istanti di tempo. Per esempio la classifica del campionato di calcio viene aggiornata solo al termine di ogni giornata di gioco. Analogamente le estrazioni del lotto avvengono ogni giorno a un determinato orario.

In pratica una grandezza discreta non ha infiniti valori in ogni istante di tempo, ma "commuta" (cambia valore) solo in corrispondenza di certi istanti ben determinati.

Grandezze digitali

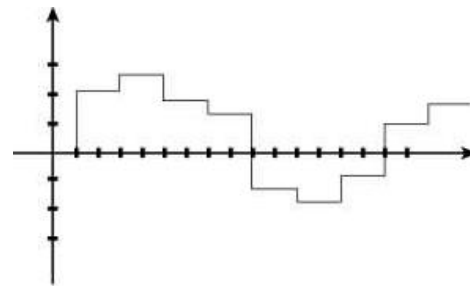
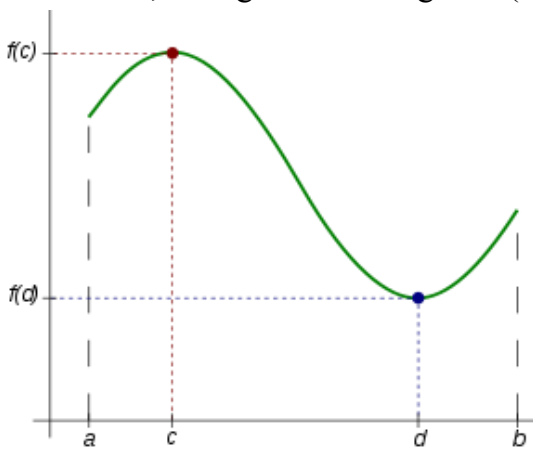
Una grandezza digitale è una grandezza discreta e quantizzata, cioè una grandezza che varia solo in corrispondenza di determinati istanti di tempo e che può assumere solo un numero finito di valori. Per esempio il contatore del tempo presente sul display di un orologio digitale procede "a scatti" di un secondo (dunque è un valore discreto) e può assumere solo valori interi (è una grandezza quantizzata).

Le grandezze digitali sono contemporaneamente discrete e quantizzate, cioè discontinue sia sull'asse delle ordinate (valori) che su quello delle ascisse (tempo).

Confronto fra grandezze analogiche e grandezze digitali

Se si rappresenta l'andamento nel tempo di una grandezza analogica (non discreta e non quantizzata), questa presenterà un grafico continuo, così come sarà continua la funzione matematica che la rappresenta.

Viceversa, una grandezza digitale (cioè discreta e quantizzata) presenta un andamento "a salti", dove l'altezza del salto minimo è costituita appunto dal quanto. Gli istanti in cui avviene il salto si dicono stanti di campionamento:



Rappresentazione digitale di grandezze analogiche

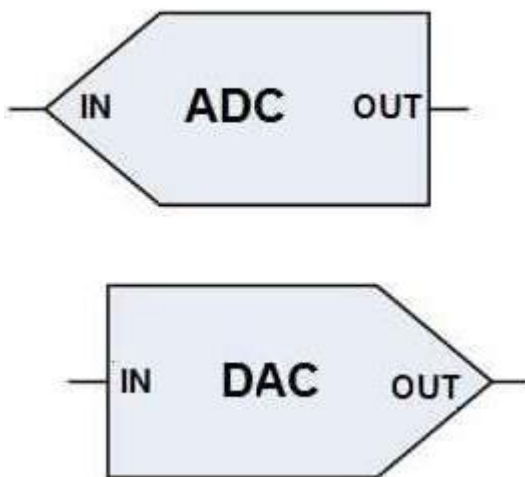
Vi sono molti casi in cui, per ragioni tecnologiche o pratiche, si vuole trattare o rappresentare una grandezza analogica, come se fosse digitale. Si pensi per esempio alla temperatura visualizzata sul display dei termometri digitali. In questo caso la temperatura è una grandezza analogica, ma essa viene digitalizzata (cioè trasformata in digitale) allo scopo di visualizzarla sul display. Uno strumento di misura simile, che però non effettua digitalizzazione della grandezza misurata, è il termometro a mercurio: in questo caso l'altezza della colonnina di mercurio è proporzionale alla temperatura misurata, senza che vi sia nessun quanto.



Nel caso del termometro digitale mostrato in figura, la temperatura è stata digitalizzata con un quanto pari a 0,1 °C.

Convertitore analogico digitale (ADC) e convertitore digitale analogico (DAC)

La digitalizzazione di una grandezza analogica è indispensabile quando si vuole elaborare la grandezza per mezzo di un calcolatore. I computer infatti sono in grado di elaborare valori numerici, ma non grandezze analogiche. Volendo per esempio registrare in un file i valori di temperatura misurati in un certo luogo, sarà necessario trasformare tali valori analogici in valori numerici che il computer è in grado di trattare. La conversione di una grandezza analogica in digitale prende il nome di conversione analogico-digitale. Non è possibile digitalizzare direttamente una qualsiasi grandezza fisica come la temperatura, ma è invece possibile digitalizzare una tensione (la quale a sua volta può essere il risultato di una misura di temperatura) e il dispositivo che effettua la conversione è detto convertitore analogico-digitale (o ADC, Analog to Digital Converter).



Viceversa accade spesso che si voglia controllare per mezzo di un computer l'andamento di una grandezza analogica. Questo è per esempio il caso di un calcolatore che regola la temperatura in un ambiente. In questo secondo caso sarà necessario trasformare i dati numerici del computer in valori analogici e si parla dunque di conversione digitale-analogica. Il dispositivo che effettua la conversione da digitale a una tensione analogica è detto convertitore digitale-analogico (o DAC, Digital to Analog Converter).

I simboli circuitali normalmente usati per rappresentare ADC e DAC sono mostrati in figura:

Si osservi che l'ingresso (IN) dell'ADC è generalmente una tensione analogica, mentre l'uscita (OUT) è di solito un valore binario a più bit. Viceversa nel caso del DAC, l'ingresso è un valore numerico binario, mentre l'uscita è una tensione analogica (o, in alcuni casi, una corrente).

Vantaggi dei segnali digitali rispetto ai segnali analogici

I segnali digitali godono di alcuni importanti pregi nei confronti di quelli analogici:

- **I segnali digitali hanno una maggiore immunità ai disturbi rispetto ai segnali analogici.** I segnali analogici sono costituiti da funzioni continue pertanto possono assumere infiniti valori: il rumore che inevitabilmente si sovrappone al segnale ha pertanto la possibilità di determinare una variazione del valore del segnale composto (segnale utile + rumore) qualunque sia la ampiezza e la potenza del rumore. I segnali digitali, invece, presentano solamente un numero finito di valori separati da una fascia "proibita". Se il rumore non ha ampiezza (e potenza) tale da determinare un superamento della fascia proibita che separa due valori contigui non si riscontra alcuna alterazione del valore.
- **I segnali digitali possono essere elaborati più facilmente dei segnali analogici**
Per elaborare matematicamente i segnali analogici si deve ricorrere agli amplificatori operazionali mediante i quale è possibile realizzare (in modo a volte molto approssimato) semplici operazioni (somma, sottrazione, logaritmo ed esponenziale, integrale e derivata rispetto al tempo, ecc.). La realizzazione di funzioni più "elaborate" può richiedere una complessità circuitale eccessiva e tale da introdurre una incertezza non accettabile per gli scopi prefissati. I segnali numerici possono invece essere elaborati mediante microprocessori e microcalcolatori i quali possono permettere la esecuzione di operazioni ed elaborazioni senza richiedere appesantimenti dell'hardware circuitale.
- **I segnali digitali possono essere registrati in maniera più fedele e stabile dei segnali analogici**
Registrare i valori di un segnale analogico, in modo da poterli recuperare in seguito, non è semplice. Ricorrendo all'uso di memorie RAM oppure di dispositivi di memoria di massa a supporto magnetico (es. hard-disk) è possibile invece registrare i segnali digitali con estrema facilità.

Volendo segnalare al contrario un possibile svantaggio del digitale rispetto all'analogico, possiamo osservare che in linea di principio il segnale analogico conserva interamente l'informazione originale, mentre la digitalizzazione (come si vedrà meglio nel seguito) comporta sempre una perdita di informazioni. Tuttavia tale perdita non ha conseguenze pratiche a condizione che si rispettino le condizioni ottimali di digitalizzazione.

Conversione analogico-digitale

La conversione analogico-digitale consiste nel trasformare un segnale analogico (cioè che varia con continuità nel tempo e può assumere infiniti valori) in una sequenza di valori numerici.



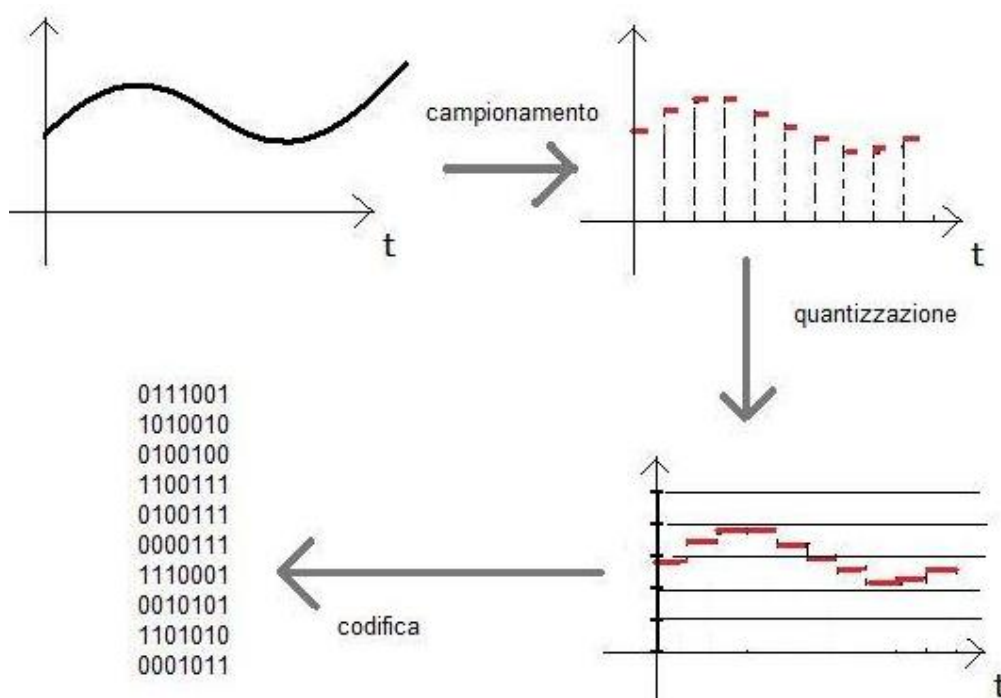
La sequenza di valori numerici prodotta in uscita dal convertitore analogico-digitale è generalmente una sequenza di numeri binari, direttamente utilizzabili da un elaboratore elettronico.

Campionamento, quantizzazione, codifica

Per effettuare correttamente la conversione da una grandezza analogica (es. una tensione) a un valore numerico digitale è necessario stabilire:

1. con quale frequenza (ovvero, ogni quanto tempo) si vuole registrare (campionare) il valore della grandezza analogica (campionamento) - questa fase corrisponde alla trasformazione della grandezza analogica in una grandezza discretizzata;
2. con quale precisione, ovvero in quanti livelli diversi si vuole suddividere i valori assunti dalla grandezza analogica (quantizzazione) - in questa fase la grandezza discretizzata viene trasformata in una grandezza quantizzata;
3. in quale modo si vuole trasformare i valori campionati e quantizzati in valori numerici (codifica).

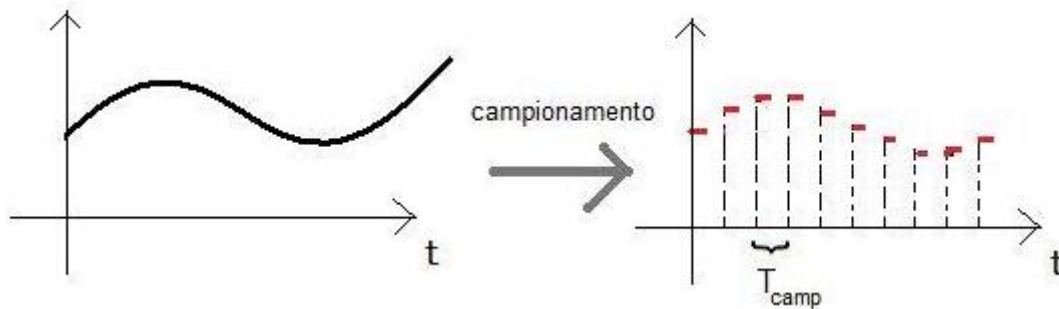
Il convertitore analogico-digitale esegue le tre operazioni precedenti in sequenza: partendo da un segnale analogico continuo, lo trasforma dapprima in un segnale sempre analogico ma campionato nel tempo; il segnale campionato viene quindi suddiviso in livelli e infine i livelli di tensione vengono convertiti in valori numerici.



Campionamento

Una grandezza analogica varia con continuità sia nel tempo che nei valori assunti. Per tornare all'esempio della temperatura, questa non solo può assumere infiniti valori, ma li può anche assumere in qualsiasi istante di tempo. Viceversa, una grandezza digitale assume solo un insieme finito di valori e li assume solo in corrispondenza di determinati istanti di tempo.

Il campionamento serve per discretizzare le variazioni nel tempo della grandezza analogica, cioè per trasformarla in una grandezza che varia solo in corrispondenza di determinati istanti di tempo. Campionare una grandezza analogica significa trasformarla in un'altra grandezza, la quale però non varia più con continuità nel tempo, ma solo in corrispondenza di determinati istanti prefissati.

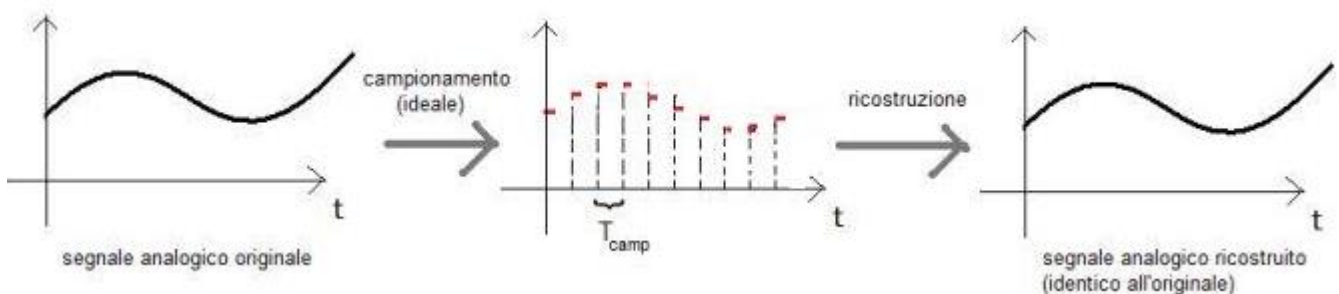


Si definisce **periodo di campionamento** (T_{camp}) il tempo che passa fra l'acquisizione di un campione e l'altro. La **frequenza di campionamento** è semplicemente il reciproco del periodo di campionamento ($f_{camp} = 1/T_{camp}$).

Intuitivamente, più breve sarà il periodo di campionamento (e più elevata, di conseguenza, la frequenza di campionamento) più il segnale campionato assomiglierà al segnale originale. Per contro tuttavia aumentando la frequenza di campionamento aumenta anche il numero di campioni (sample) che bisogna acquisire e occorrerà anche utilizzare dispositivi di campionamento più veloci.

La scelta del periodo di campionamento dovrà dunque essere in generale un compromesso fra la necessità di riprodurre il più fedelmente possibile il segnale e i problemi tecnologici derivanti dal numero eccessivo di campioni da trattare e dalla rapidità del campionamento stesso.

In generale il campionamento di una grandezza analogica è ottimale se non comporta perdita di informazioni, ovvero se è possibile ricostruire perfettamente la grandezza analogica originaria a partire dai suoi campioni.



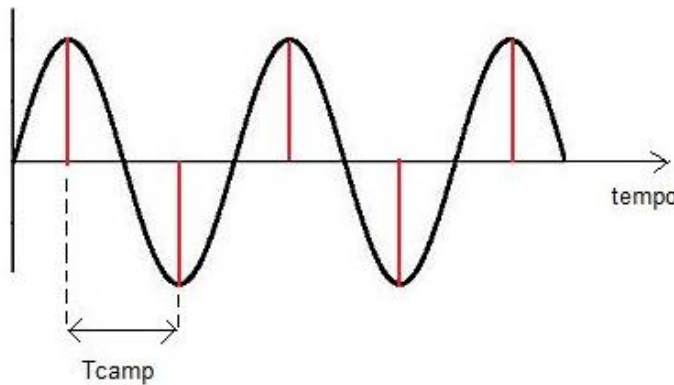
E' possibile dimostrare che la ricostruzione perfetta (*senza perdita di informazioni*) di un segnale analogico a partire dai suoi campioni è possibile solo nel caso in cui il segnale di partenza ha una banda limitata.

Teorema del campionamento (di Nyquist-Shannon) applicato a una sinusoidale

Il **teorema del campionamento** (o teorema di Nyquist-Shannon) afferma che, per campionare correttamente (senza perdita di informazioni) un segnale a banda limitata, è sufficiente **campionarlo con una frequenza di campionamento pari almeno al doppio della massima frequenza del segnale** (tale frequenza viene anche detta **frequenza di Nyquist**).

$$f_c \geq 2F_{max}$$

Facciamo un esempio semplice. Supponiamo di voler campionare un segnale sinusoidale con $f = 1 \text{ kHz}$ e periodo $T = 1/f = 1 \text{ ms}$. Il teorema del campionamento afferma che per



campionare correttamente tale segnale occorre usare una frequenza di campionamento almeno doppia della frequenza di segnale 1 kHz . Dunque $f_{camp} > 2 \text{ kHz}$ e $T_{camp} < 0.5 \text{ ms}$. Come si può notare, questo valore corrisponde a campionare il segnale due volte per ogni periodo, come mostrato in figura:

Può sembrare sorprendente che occorran così pochi campioni per rappresentare correttamente una sinusoidale. In realtà il risultato del teorema di Shannon, così come è stato esposto, è puramente teorico, per diverse ragioni:

- la condizione $f_{camp} = 2 f_{segnale}$ vale solo se il segnale è rigorosamente a banda limitata, cioè se è possibile individuare nel suo spettro una frequenza massima; a parte le sinusoidi, nessun segnale reale di interesse pratico ha una banda limitata.
- il campionamento dev'essere effettuato in modo sincrono con i massimi e i minimi del segnale sinusoidale; se i campioni non sono esattamente sincronizzati con le variazioni del segnale, la ricostruzione del segnale non è fedele.
- per ricostruire il segnale sinusoidale a partire dai suoi (pochi) campioni, occorre avere a disposizione un filtro passa-basso ideale, in grado di eliminare dal segnale campionato tutte le armoniche con frequenza superiore a $f_{segnale}$ e a far passare tutte le altre senza attenuazione; un filtro del genere non è realizzabile nella pratica.

Es. per un segnale compreso nella banda audio $f=20\text{Hz} - 20.000\text{Hz}$ si deve campionare ad una frequenza minima di 40.000Hz

Per segnali lentamente variabili nel tempo, grandezze ambientali, (temperatura, umidità, pressione atmosferica, ecc), non è possibile stabilire una frequenza del segnale.

In questo caso si esegue la **quantizzazione** e la **codifica**.

Il convertitore interno di Arduino esegue la quantizzazione e la codifica.

Convertitore A/D interno

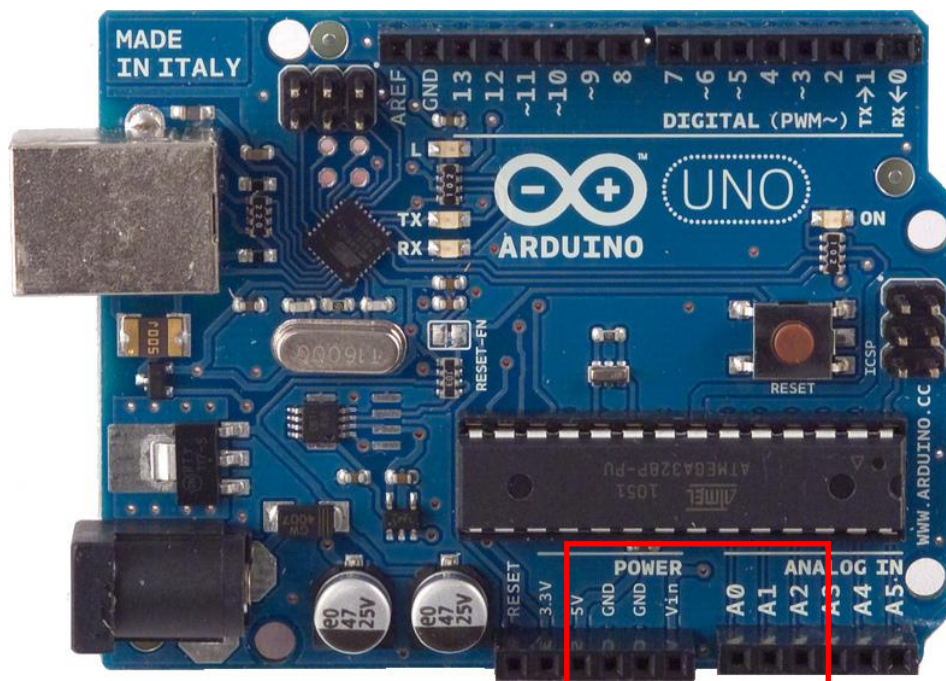
Arduino dispone al proprio interno un convertitore Analogico/Digitale a 10 bit a 6 canali (A0 – A5) che consente di collegare direttamente al chip dispositivi esterni, i quali generano segnali che abbiano un'escursione di tensione compresa tra 0V÷5V.

Il segnale analogico viene letto attraverso un piedino di I/O abilitato al funzionamento analogico, convertito in un numero digitale a 10 bit e memorizzato all'interno della memoria RAM (Variabile/Array).

Il convertitore è caratterizzato dai seguenti parametri:

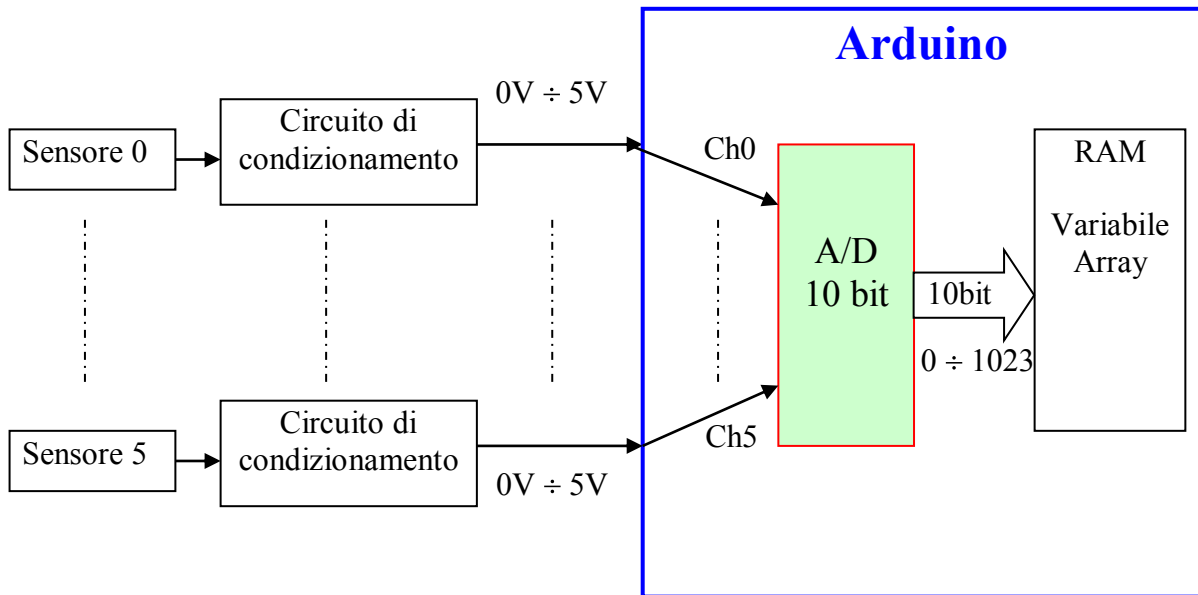
- 1) Risoluzione digitale 10 bit
- 2) VFS (tensione fondo scala)=5V
- 3) Range di Input=0V÷5V
- 4) Range di output=0÷1023
- 5) Risoluzione analogica $q = \frac{Vfs}{2^{10}} = \frac{5-0}{1024} = 4,88mV$

In figura sono evidenziati i pin abilitati al funzionamento analogico.



In tabella è riportata la corrispondenza canale-porta-pin

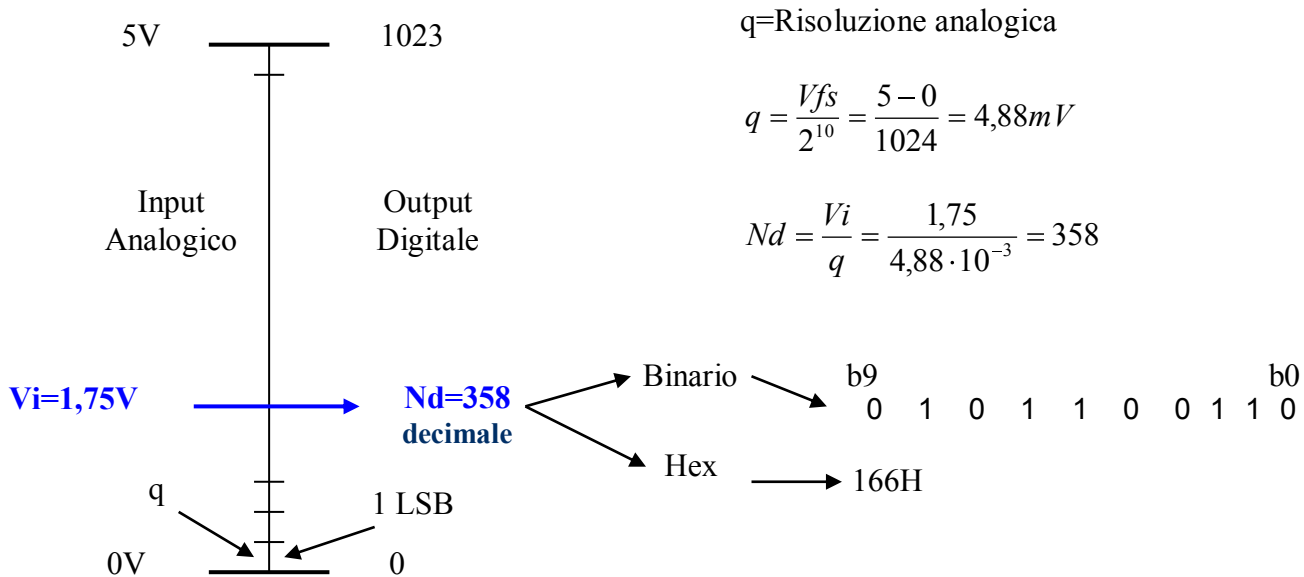
Canale	Arduino Uno		
	Porta		
A/D Channel 0	A0		
A/D Channel 1	A1		
A/D Channel 2	A2		
A/D Channel 3	A3		
A/D Channel 4	A4		
A/D Channel 5	A5		



In figura è riportato lo schema a blocchi di un sistema in grado di acquisire, convertire e memorizzare 6 grandezze analogiche senza l'ausilio di un multiplexer (Selettore) esterno. Se i segnali sono superiori a 6 si deve inserire all'esterno di Arduino un multiplexer (Selettore), nelle prossime pagine verrà riportata la struttura di questo sistema.

In figura un esempio di conversione.

Il valore analogico $V_i=1,75V$ viene convertito nel numero binario 0101100110 corrispondente al numero decimale 358 oppure 166H.



Comandi

Il convertitore A/D interno viene gestito tramite il comando AnalogRead.

Sintassi di AnalogRead

Variabile=analogRead(Canale)

La variabile deve essere di tipo Int, mentre il canale può andare da 0 a 5 (A0 – A5)

Esempio:



```
File Modifica Sketch Strumenti Aiuto
Esempio_convertitore
1 byte ch=0; // Variabile canale A0 del Conv. A/D
2 int numero=0; // Variabile risultato conversione
3 float volt=0; // Variabile per conversione numero in volt
4
5 void setup()
6 {
7   Serial.begin(9600); // setup serial
8 }
9
10 void loop()
11 {
12   numero=analogRead(ch) ;Serial.println(numero);
13   volt=numero*5.0/1024.0;Serial.println(volt);
14 }
```

Esercizio 24 – Rilevazione temperatura con il Sensore TMP 36 e Monitor Seriale

Dimensionare e programmare un circuito in grado di rilevare la temperatura ambiente con il sensore TMP36 e visualizzare il valore tramite il Monitor Seriale dell'IDE

1. Disegnare lo schema elettrico
2. Disegnare lo schema di cablaggio
3. Scrivere il programma

Soluzione

Il TMP36 è un sensore di temperatura analogico, fornisce un'uscita lineare pari a 10mV/°C con un offset di 500mV nel range -40°C +125°C, alla temperatura di 0°C il sensore eroga una tensione di **500mV**.

Il legame Temperatura misurata (°C) e tensione di uscita è dato dalla seguente caratteristica di trasferimento:

$V_{out}=K \cdot T+0,5$ dove $K=10\text{mV}/^{\circ}\text{C}$, $T=\text{temperatura espressa in }^{\circ}\text{C}$ e 0,5 rappresenta l'Offset.

Trasformazione in:

$T(^{\circ}\text{F})=1,8 \cdot T+32$

$T(\text{K})=273,15+T$

In figura 1 Grafico (*Vout in funzione della Temperatura*) e Caratteristiche tecniche.

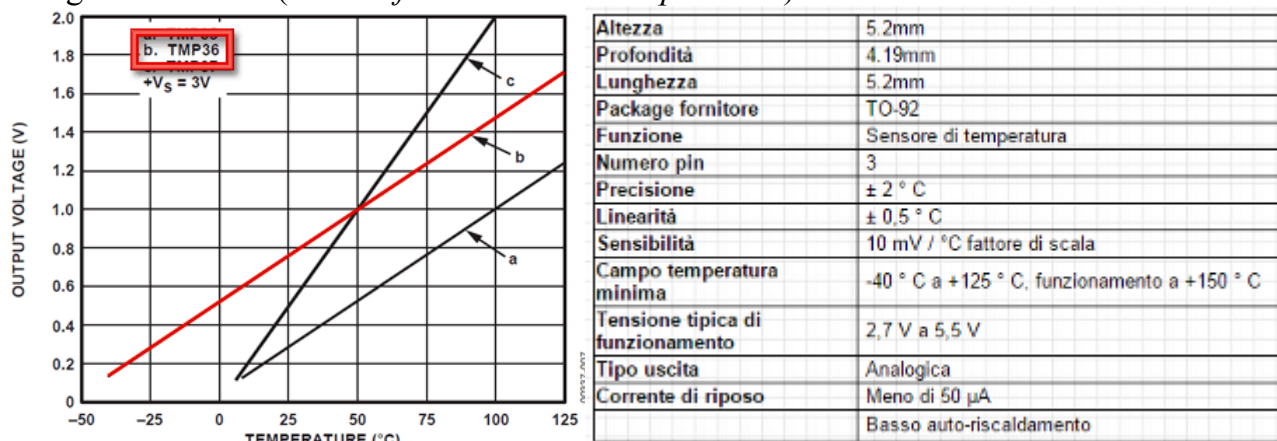


Fig. 1

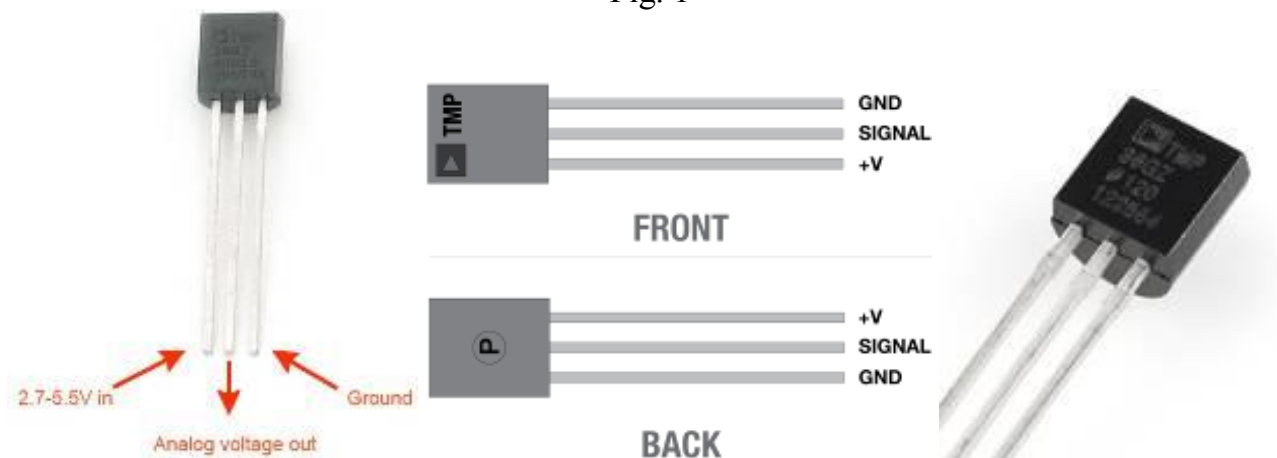


Fig. 2 (Piedinatura e foto)

In Fig. 3 è riportato lo schema elettrico, in fig. 4 lo schema di cablaggio con Fritzing.

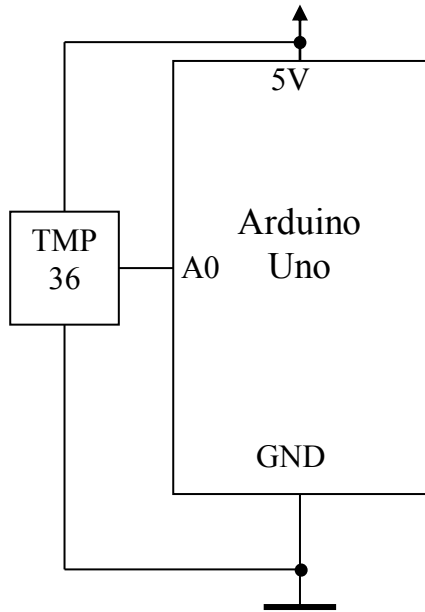


Fig. 3 – Schema elettrico

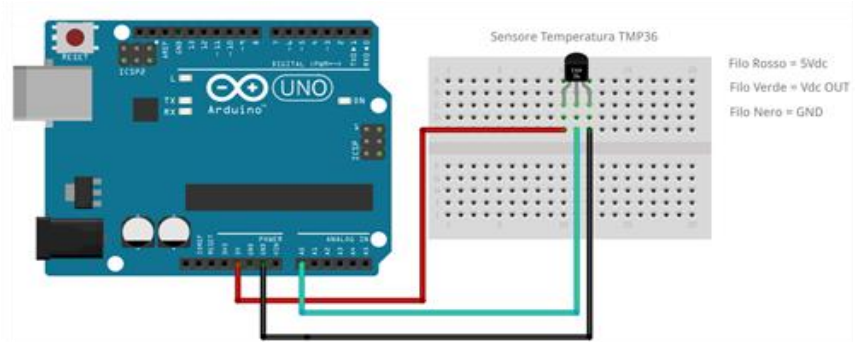


Fig. 4: Cablaggio con Fritzing

Il segnale analogico fornito dal sensore (V_{out}) viene inviato al convertitore A/D interno (Pin A0) che lo trasforma in un numero nel range (0÷1023).

Tramite software si trasforma il numero nel valore di Tensione e nel valore di Temperatura .

Formule:

$V_{out} = K \cdot T + 0,5$ [V] dove $K = 10\text{mv}/^\circ\text{C}$, $T = \text{temperatura espressa in } ^\circ\text{C}$ e 0,5 rappresenta l'Offset.

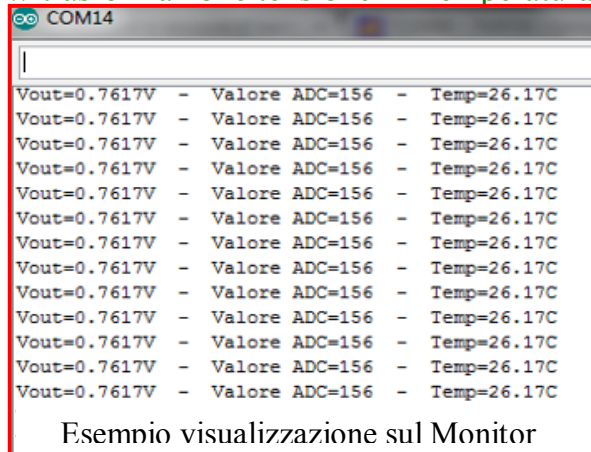
$$N = \frac{V_{out}}{5} \cdot 1024 \quad [N. \text{ out A/D}] \qquad V_{out} = \frac{N \cdot 5}{1024} \quad [V] \qquad T = \frac{V_{out} - 0,5}{K} \quad [^\circ\text{C}]$$

Programma

// **Esercizio 24 - Rilevazione temperatura con il Sensore TMP 36 e Monitor Seriale**

```
void setup()
{
  Serial.begin(9600); // Inizializzazione monitor seriale
}
void loop()
{
  int valoreADC = analogRead(0);
  float voltage = (valoreADC/1024.0)* 5.0;
  float temp=(voltage-0.5)/0.01;
  // Invio dati al PC tramite seriale
  Serial.print("Vout=");
  Serial.print(voltage,4);
  Serial.print("V - Valore ADC=");
  Serial.print(valoreADC);
  Serial.print(" - Temp=");
  Serial.print(temp,2);
  Serial.println("C");
  delay (1000);
}
```

// Conv A/D di Vout del sensore TMP36
 // trasformazione valore ADC in Tensione
 // trasformazione tensione in Temperatura



Esemplificazione visualizzazione sul Monitor

Esercizio 25 – Rilevazione temperatura con il Sensore TMP 36 e Display LCD 16x2

Dimensionare e programmare un circuito in grado di rilevare la temperatura ambiente con il sensore TMP36 e visualizzare il valore tramite il Display LCD 16x2.

1. Disegnare lo schema elettrico
2. Disegnare lo schema di cablaggio
3. Scrivere il programma

Soluzione

In Fig. 1 è riportato lo schema elettrico.

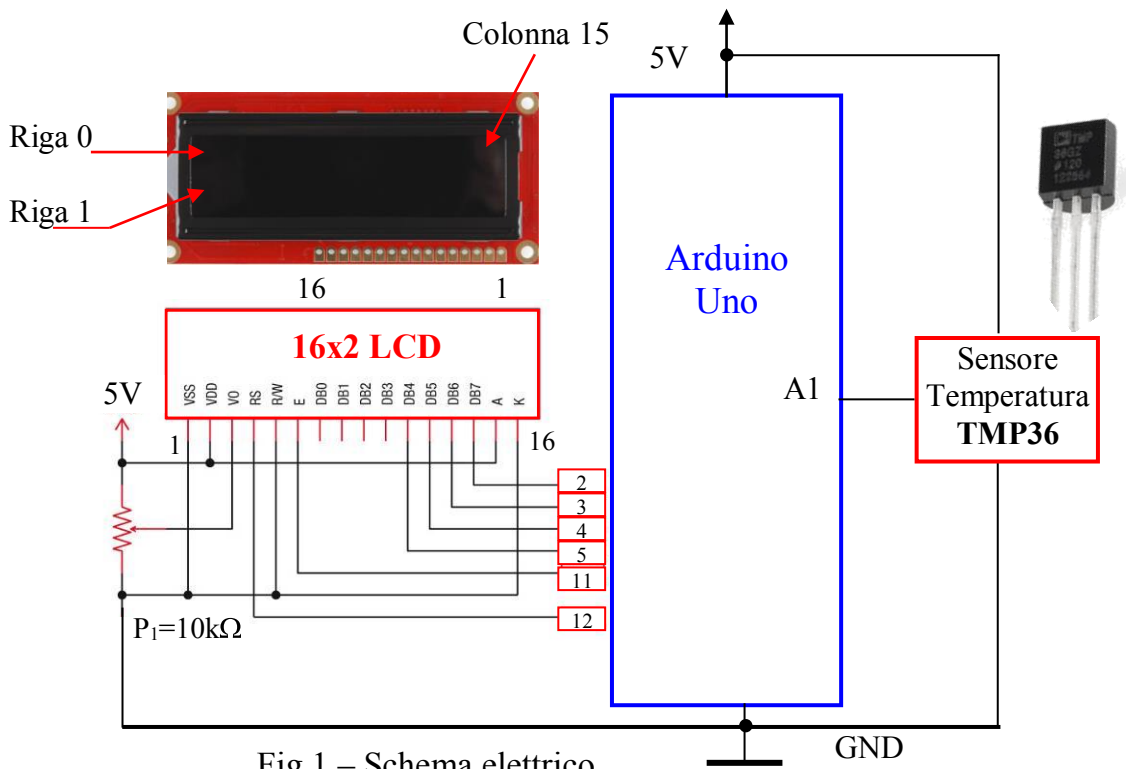


Fig.1 – Schema elettrico

Programma

```
/* Esercizio 25 - Rilevazione temperatura con il Sensore TMP 36 e Display LCD 16x2
```

```
Descrizione pin:
```

```
Pin +5V      -> Alimentazione
```

```
Pin GND      -> Alimentazione
```

```
Pin Digital 2 -> Bus DB7 dati LCD
```

```
Pin Digital 3 -> Bus DB6 dati LCD
```

```
Pin Digital 4 -> Bus DB5 dati LCD
```

```
Pin Digital 5 -> Bus DB4 dati LCD
```

```
Pin Digital 11 -> terminale E display LCD
```

```
Pin Digital 12 -> terminale RS display LCD
```

```
Pin Analogico A1 -> lettura sensore TMP36
```

```
*/
```

```
#include <LiquidCrystal.h> // Libreria gestione Display LCD
```

```
// Selezione dei pin utilizzati dal display LCD
```

```
// Sintassi LiquidCrystal(rs, enable, d4, d5, d6, d7)
```

```
LiquidCrystal lcd(12,11,5,4,3,2);
```

```
void setup()
```

```
{
```

```
lcd.begin (16,2); // Inizializzazione libreria per display LCD 16x2
```

```
}
```

```
void loop()
```

```
{
```

```
int valoreADC = analogRead(1);
```

```
float voltage = (valoreADC/1024.0)* 5.0;
```

```
float temp=(voltage-0.5)/0.01;
```

```
// Invio dati al Display LCD
```

```
lcd.clear(); // Pulisce lo schermo LCD
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Vout=");
```

```
lcd.setCursor(5, 0);
```

```
lcd.print(voltage,3);
```

```
lcd.setCursor(10, 0);
```

```
lcd.print("V");
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("Temp=");
```

```
lcd.print(temp,2);
```

```
lcd.setCursor(10, 1);
```

```
lcd.print("C");
```

```
delay (2000); //Pausa di 2 secondi
```

```
}
```

Esercizio 28 – Controllo Servomotore con un potenziometro

Soluzione

In Fig. 1 è riportato lo schema elettrico, in fig. 2 lo schema di cablaggio con Fritzing.

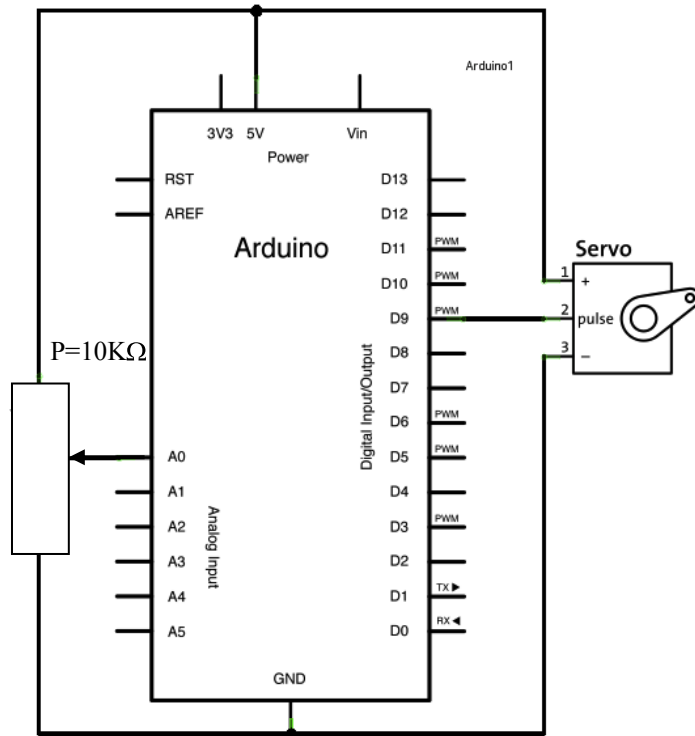


Fig.1 – Schema elettrico

Programma

// Esercizio 28: Controllo Servomotore con un potenziometro

#include <Servo.h> // richiama la libreria di gestione dei servomotori

Servo myservo; // crea il servo oggetto “myservo”

int pos = 0; // variabile per la memorizzazione della posizione (in gradi angolari) del perno

void setup()

{
Serial.begin(9600);

myservo.attach(9); // assegna il servo oggetto “myservo” alla porta 9

myservo.write(0); // sposta il perno nella posizione iniziale

}

void loop()

{

pos=map(analogRead(0), 0, 1023, 0, 180); // legge il valore sulla porta A0
// e mappa il valore nell’intervallo 0° ± 180°

myservo.write(pos); // sposta il perno nella posizione memorizzata in “pos”

delay(15);

}

Esercizio 29 – Inseguitore solare con Fotoresistenze e Servomotore

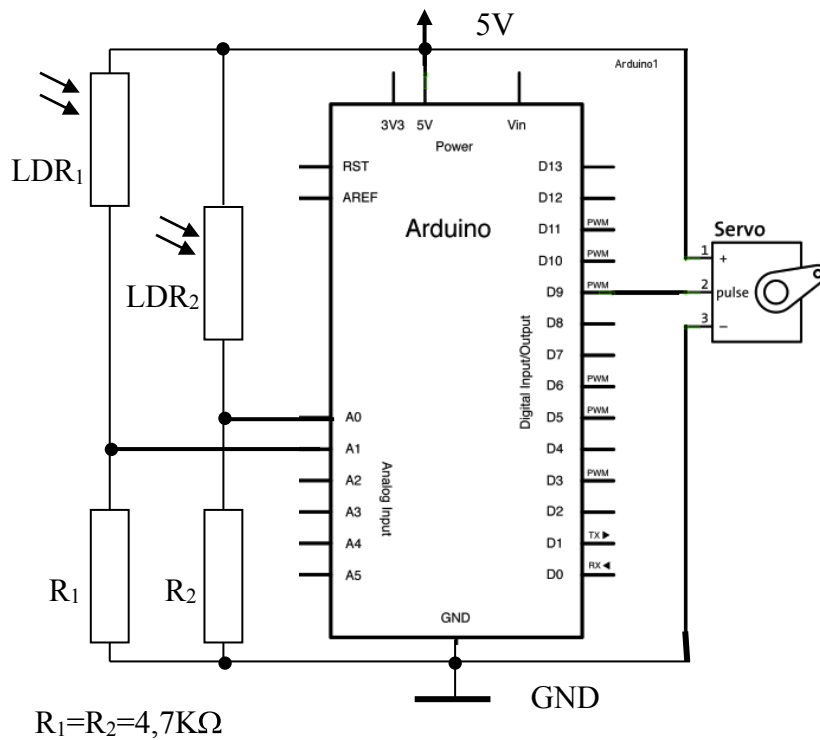


Fig. 1: Schema elettrico

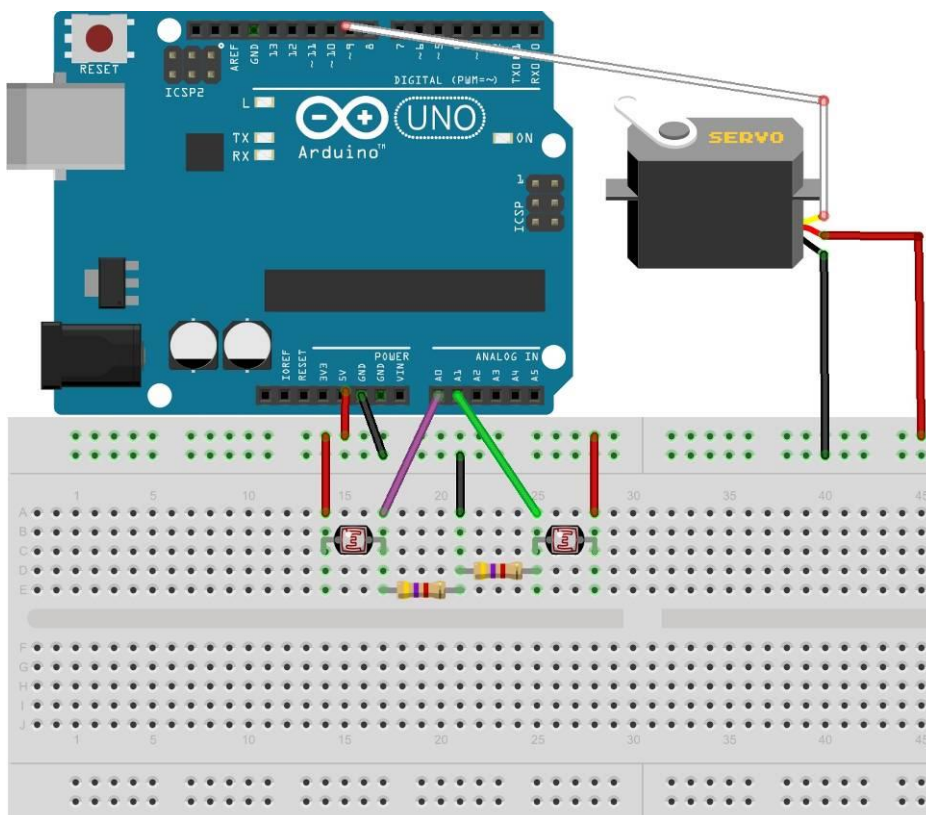


Fig. 2: Cablaggio con Fritzing

Programma

// **Esercizio 29: Inseguitore solare con Fotoresistenze e Servomotore**

#include <Servo.h> // richiama la libreria di gestione dei servomotori

Servo myservo; // crea il servo oggetto “myservo”

int Valore;

int Centro=90; // Centro del servo

```
void setup() {  
  Serial.begin(9600);  
  myservo.attach(9); // Collega il servo al PIN 9  
}
```

```
void loop() {  
  int sensorValue = analogRead(A0); // Legge il valore del sensore LDR1 di sinistra  
  int sensorValue2 = analogRead(A1); // Legge il valore del sensore LDR2 di destra
```

```
  Valore=(sensorValue-sensorValue2)/10;
```

```
  /*
```

```
  Calcola la differenza tra i due sensori e la divide per 10,  
  in questo modo una differenza di 10 equivale a 1, una differenza  
  di 53 equivale a 5 e così via
```

```
  */
```

```
  if (Valore==0) myservo.detach(); else myservo.attach(9);
```

```
  /*
```

```
  Se la differenza è uguale a 0 scollega il servo altrimenti lo  
  collega (serve per risparmiare energia)
```

```
  */
```

```
  if (Valore>30) Valore=30;
```

```
  if (Valore<-30) Valore=-30;
```

```
  // Limita la differenza massima tra -30 e + 30
```

```
  Serial.println(Valore); // Visualizza il valore sulla seriale
```

```
  myservo.write(Centro+Valore);
```

```
  /*
```

```
  Sposta il perno nella posizione Centro+Valore - Max=90+30=120; Min=90-30=60
```

```
  Se “Valore” è positivo ruota il perno a DX,
```

```
  Se “Valore” è negativo ruota il perni a SX.
```

```
  La velocità è direttamente proporzionale al valore assoluto di Valore
```

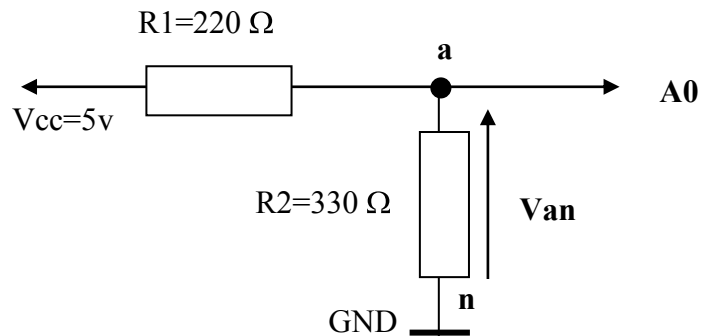
```
  */
```

```
  delay(15);
```

```
}
```


 **Esercizio 42 - Simulazione Legge di Ohm**

a) Arduino Ohm 1



Programma

```

/* -----
- Arduino Ohm 1:Partitore con 2 Resistori (R1, R2)
-- Calcolo: RTot, Corrente, Caduta di tensione ai capi di R2
-- Potenza Erogata da Vcc, potenza dissipata sui singoli Resistori
---- IPSIA - Antonio Guastaferrò - San Benedetto del Tronto
---- Vers 1.0 - A.S.2017-2018 - CL:4A_IPAI
-----
*/

float Van,I,Ima,Pr1,Pr2,PE,Rt;
float R1=220,R2=330;
float Vcc=5.0;

void setup()
{
  Serial.begin(9600); // Inizializzazione monitor seriale
}

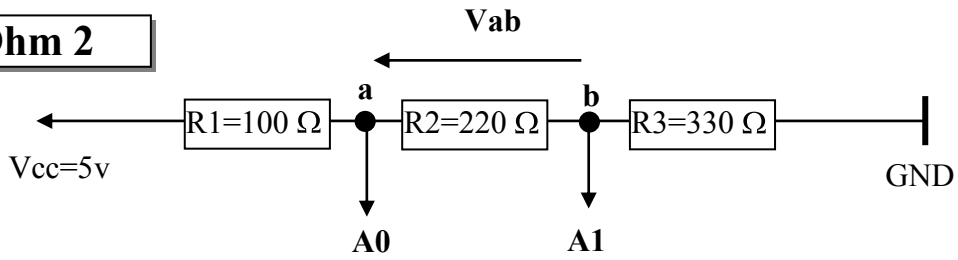
void loop()
{
  Rt=R1+R2;
  Van=5.0*analogRead(A0)/1024.0;

  I=Van/R2;Ima=1000.0*I;
  Pr1=1000.0*R1*I*I;Pr2=1000.0*R2*I*I;PE=Vcc*Ima;
  // Invio dati al PC tramite seriale
  Serial.print("Rt=");Serial.print(Rt,0);Serial.print(" Ohm");
  Serial.print(" Van=");Serial.print(Van,3);Serial.print("V");
  Serial.print(" I=");Serial.print(I,5);Serial.print("A");
  Serial.print(" Ima=");Serial.print(Ima,2);Serial.print("mA");
  Serial.print(" Pr1=");Serial.print(Pr1,2);Serial.print("mW");
  Serial.print(" Pr2=");Serial.print(Pr2,2);Serial.print("mW");
  Serial.print(" PE =");Serial.print(PE,2);Serial.println("mW");
  delay (5000);
}

```

b) Arduino Ohm 2

Circuito



Programma

```

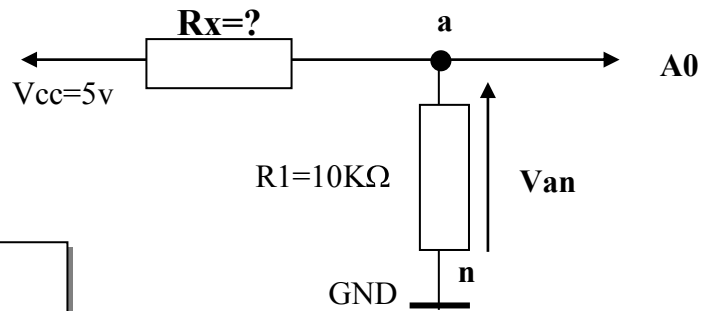
/* -----
- Arduino Ohm 2:Partitore con 3 Resistori (R1, R2, R3)
-- Calcolo: RTot, Corrente, Caduta di tensione ai capi di R2
-- Potenza Erogata da Vcc, potenza dissipata sui singoli Resistori
---- IPSIA - Antonio Guastaferrò - San Benedetto del Tronto
---- Vers 1.0 - A.S.2017-2018 - CL:4A_IPAI
-----
*/
float Va,Vb,Vab,I,Ima,Pr1,Pr2,Pr3,PE,Rt;
float R1=100,R2=220,R3=330;
float Vcc=5.0;
void setup()
{
  Serial.begin(9600); // Inizializzazione monitor seriale
}
void loop()
{
  Rt=R1+R2+R3;
  Va=5.0*analogRead(A0)/1024.0;
  Vb=5.0*analogRead(A1)/1024.0;
  Vab=(Va-Vb); I=Vab/R2;Ima=1000.0*I;
  Pr1=1000.0*R1*I*I;Pr2=1000.0*R2*I*I;Pr3=1000.0*R3*I*I;PE=Vcc*Ima;
  // Invio dati al PC tramite seriale
  Serial.print("Rt=");Serial.print(Rt,0);Serial.print(" Ohm ");
  Serial.print(" Vab=");Serial.print(Vab,3);Serial.print(" V ");
  Serial.print(" I=");Serial.print(I,5);Serial.print(" A");
  Serial.print(" Ima=");Serial.print(Ima,2);Serial.print(" mA ");
  Serial.print(" Pr1=");Serial.print(Pr1,2);Serial.print(" mW ");
  Serial.print(" Pr2=");Serial.print(Pr2,2);Serial.print(" mW ");
  Serial.print(" Pr3=");Serial.print(Pr3,2);Serial.print(" mW ");
  Serial.print(" PE =");Serial.print(PE,2);Serial.println(" mW ");
  delay (5000);
}

```

Risultato

Rt=650 Ohm	Vab=1.689V	I=0.00768A	Ima=7.68mA	Pr1=5.90mW	Pr2=12.97mW	Pr3=19.46mW	PE =38.40mW
Rt=650 Ohm	Vab=1.689V	I=0.00768A	Ima=7.68mA	Pr1=5.90mW	Pr2=12.97mW	Pr3=19.46mW	PE =38.40mW
Rt=650 Ohm	Vab=1.689V	I=0.00768A	Ima=7.68mA	Pr1=5.90mW	Pr2=12.97mW	Pr3=19.46mW	PE =38.40mW
Rt=650 Ohm	Vab=1.694V	I=0.00770A	Ima=7.70mA	Pr1=5.93mW	Pr2=13.05mW	Pr3=19.57mW	PE =38.51mW
Rt=650 Ohm	Vab=1.689V	I=0.00768A	Ima=7.68mA	Pr1=5.90mW	Pr2=12.97mW	Pr3=19.46mW	PE =38.40mW

c) Arduino Ohm 3 - Ohmmetro oppure Ohmetro



Range ottimale: 50Ω ÷ 120KΩ

```

/* -----
- Arduino Ohm 3:Ohmetro
-- Lettura valore Resistore Rx
--
---- IPSIA - Antonio Guastaferrò - San Benedetto del Tronto
---- Vers 1.0 - A.S.2017-2018 - CL:4A_IPAI
-----
*/
float Rx, Van, I;
float R1=10000;
float Vcc=5.0;

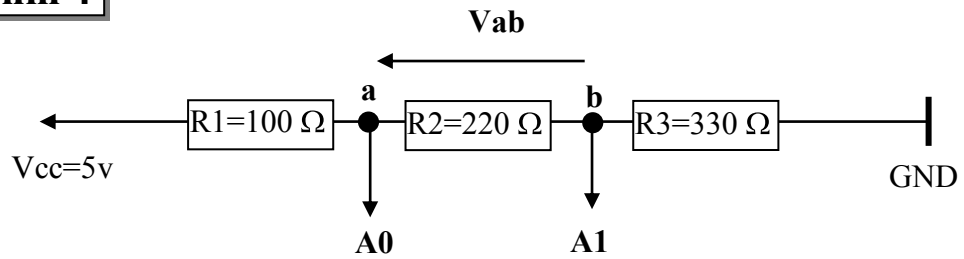
void setup()
{
  Serial.begin(9600); // Inizializzazione monitor seriale
}

void loop()
{
  Van=5.0*analogRead(A0)/1024.0;
  I=Van/R1;
  Rx=(Vcc-Van)/I;
  // Invio dati al PC tramite seriale
  Serial.print("Rx=");Serial.print(Rx,0);Serial.println(" Ohm");
  delay (5000);
}

```

d) Arduino Ohm 4

Circuito



Programma

```

/* -----
- Arduino Ohm 4: Struttura a ponte con 4 resistori
-- Determinazione Vab (Diagonale del Ponte)
---- IPSIA - Antonio Guastaferrò - San Benedetto del Tronto
---- Vers 1.0 - A.S.2017-2018 - CL:4A_IPAI
-----
*/
float Va,Vb,Vab;
void setup()
{
  Serial.begin(9600); // Inizializzazione monitor seriale
}
void loop()
{
  Va=5.0*analogRead(A0)/1024.0;
  Vb=5.0*analogRead(A1)/1024.0;
  Vab=(Va-Vb);

  // Invio dati al PC tramite seriale

  Serial.print(" Vab=");Serial.print(Vab,3);Serial.print("V ");
  delay (5000);
}

```