

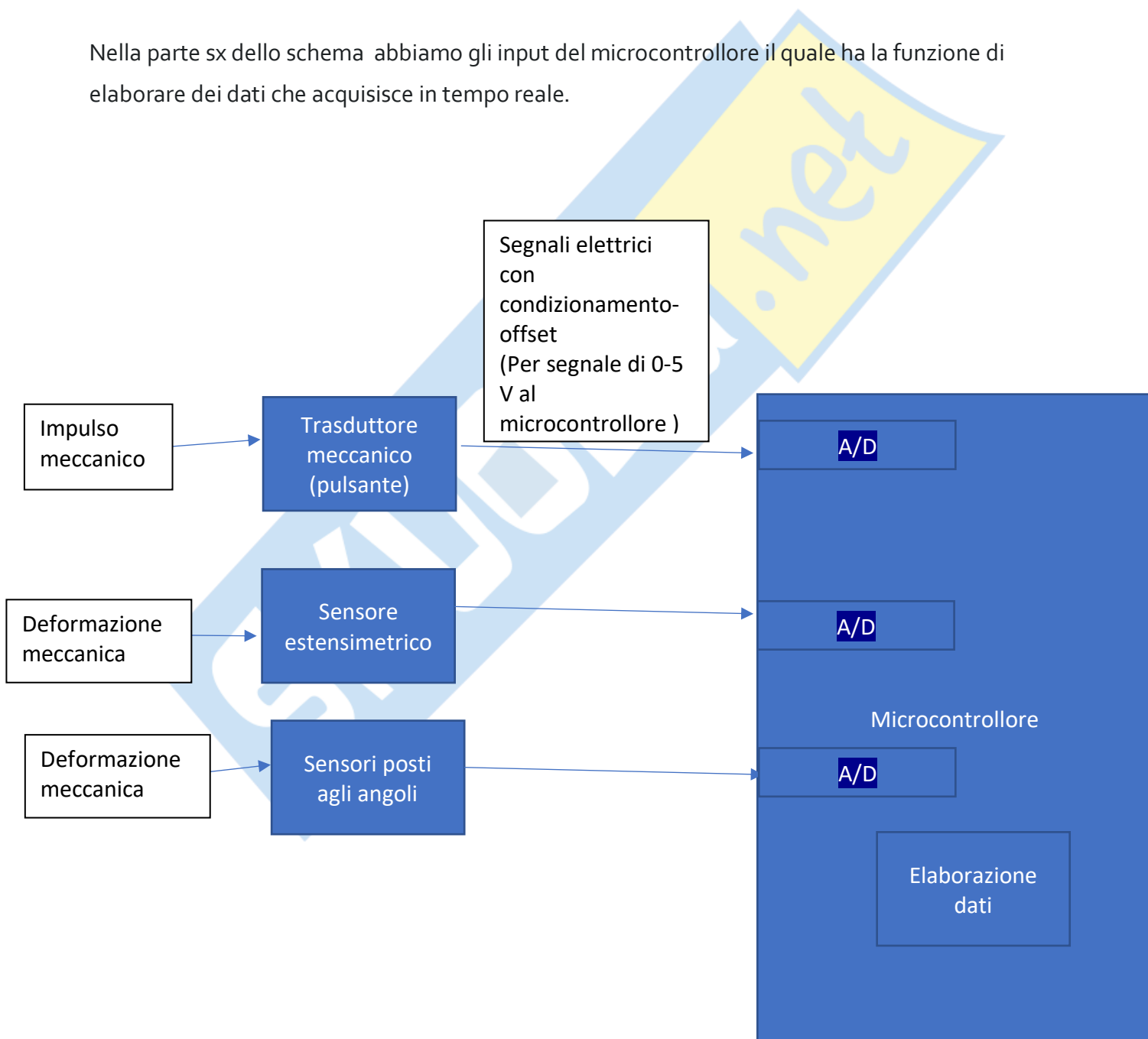
**PROPOSTA DI SOLUZIONE PER LA SECONDA PROVA DI MATURITÀ 2018**

**TRACCIA:** Tecnologie e Progettazione dei sistemi elettrici ed elettronici

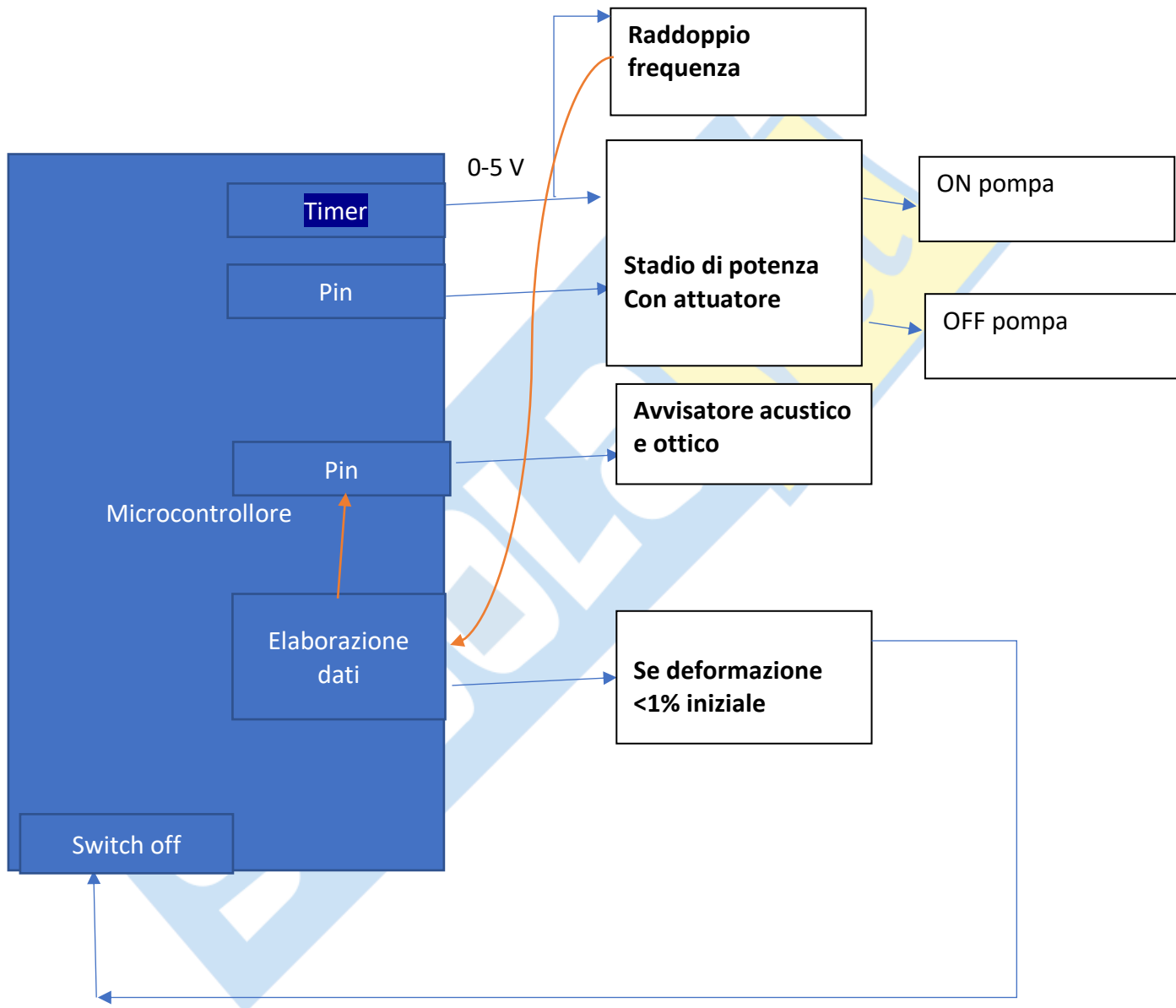
**ARGOMENTO:** Acquisizione dati in tempo reale con conseguenti azionamenti

**PUNTO 1)**

Nella parte sx dello schema abbiamo gli input del microcontrollore il quale ha la funzione di elaborare dei dati che acquisisce in tempo reale.



Nella parte dx abbiamo gli output che vengono determinati dall'elaborazione dati in base agli input in tempo reale, quando il timer integrato nel microcontrollore esaurisce il suo contatore fornisce un' uscita che va ad azionare l'attuatore della pompa.



**PUNTO 2)**

Data la frequenza del segnale derivante dai sensori massima pari a 15 Hz , si assume una frequenza di campionamento pari a 5 volte , quindi **75 Hz** in modo da evitare fenomeni di aliasing e fenomeni legati al sample and hold del convertitore.

Siccome vi sono 5 canali da cui vengono acquisiti i dati ( il sensore centrale più i quattro agli angoli), e il campionatore è solo uno, la frequenza di campionamento sarà 5 volte quella di sopra.

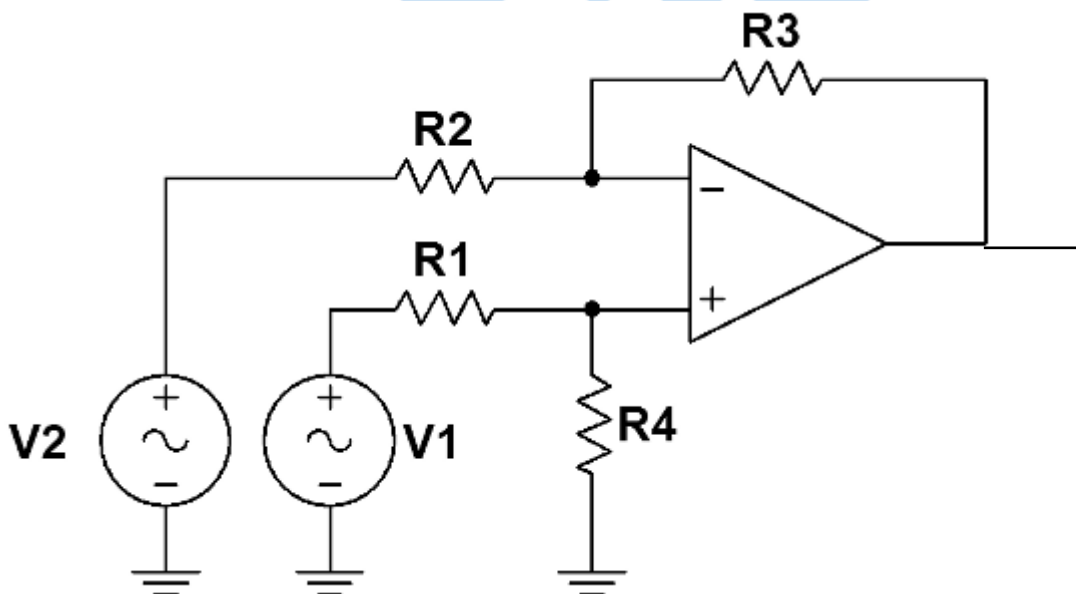
Per cui l'intervallo di campionamento sarà pari a  $1/(75*5)$  s.

E' accettabile in ogni caso, per un canale una frequenza  $>2 f_{max}$

E' stata presa la decisione più stringente, per gli altri sensori ( più statici) la frequenza non sarà un problema per cui il campionamento scelto è opportuno anche questi ultimi.

**PUNTO 3)**

L'unico **sensor** per cui abbiamo i dati è quello nel punto e) , esso dà una tensione differenziale di 0-24 V, siccome l'entrata del microcontrollore è solo una , è necessario utilizzare un amplificatore operazionale differenziale in figura:



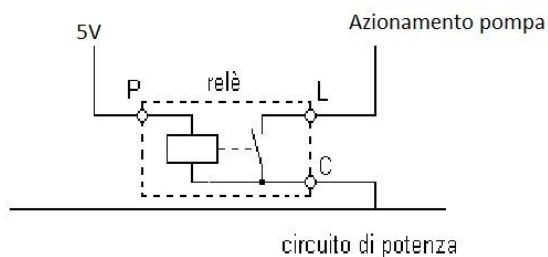
La tensione  $V_{out}$  ( nell'estremità dx dell'operazionale ) vale:

$$V_{out} = (V_1 - V_2) * R_3 / R_2$$

La nostra differenza massima  $V_1 - V_2$  vale 0-24 V che dobbiamo trasformarla in 0-5 V per l'ingresso al microcontrollore , abbiamo che :

$5 = 24 * R_3 / R_2 \rightarrow R_3 / R_2 = 5 / 24$  , per cui , teoricamente  $R_3$  dovrebbe valere 5 k $\Omega$  e  $R_2 = 24$  k $\Omega$  ,se non sono reperibili in commercio, è possibile metterne una reperibile e una variabile in serie in modo da avere esattamente quel rapporto . Si sono scelte resistenze nell'ordine dei k $\Omega$  per limitare la corrente ed evitare di bruciare l'operazionale.

Per quanto riguarda gli **attuatori** , il loro segnale in ingresso non può essere di 5 V come il segnale in uscita dal microcontrollore, per cui si deve progettare uno stadio di potenza , si utilizza un relè con segnale di commutazione di 5 V come in figura.



#### PUNTO 4)

Per il tipo di microcontrollore si assume Arduino

La funzione millis() restituisce il numero di millisecondi da quando la scheda Arduino ha iniziato l'esecuzione del programma

TIMEOUT è una costante e vale 120'000 ms in quanto corrisponde a 120 secondi dopo il quale deve essere azionata la pompa.

Lo stato del valore della deformazione viene letto dal pin analogico Ao.

E' necessario fissare la variabile STARTING\_TIME=0 una volta attuata la pompa altrimenti nel ciclo successivo ON pompa e OFF pompa vanno in conflitto.

E' stato deciso che una volta che il segnale supera 4,9 V ( deformazione massima) si fanno le relative attuazioni, in quanto se si pone la condizione con esattamente uguale a 5 per eventuali errori non potrebbe arrivare a 5 non permettendo, così , al programma, di funzionare.

La valvola di apertura , una volta aperta, deve richiudersi per potersi riaprire al ciclo successivo , per cui , quando l'estensimetro va in rilassamento (if (valore\_deformazione< 1V)), ( si poteva scegliere anche un valore diverso da 1 V , l'importante è che fosse più piccolo di 5 V) comandiamo la chiusura della valvola.

```
*****  
*****
```

```
int pin_attuazione_pompa=7;
```

```
int pin_valvola_apertura =6;
```

```

int pin_schiacciamento pulsante =5;

unsigned long STARTING_TIME;

const unsigned long TIMEOUT = 120000;

void setup ()
{
    pinMode( pin_attuazione_pompa, OUTPUT)
    pinMode(pin_valvola_apertura, OUTPUT)
    pinMode(pin_schiacciamento_pulsante,INPUT)
}

void loop ()
{
    float valore_deformazione;
    float volt;

    if ( digitalRead(pin_schiacciamento_pulsante)==1)
    {
        STARTING_TIME= millis();
    }

    if ( (millis()-STARTING_TIME )> TIMEOUT)
    {
        digitalWrite(pin_attuazione_pompa, HIGH);
        STARTING_TIME=0;
    }

    valore_deformazione = analogRead(0);
    volt = valore_deformazione/1024*5; // conversione in volt del valore digitale ( infatti nel //
//processore entrano variabili in formato digitale)

```

```
if (valore_deformazione > 4.9 V)
{
    digitalWrite(pin_attuazione_pompa, LOW);
    digitalWrite(pin_valvola_apertura, HIGH);
}
if (valore_deformazione < 1V)
{
    digitalWrite(pin_valvola_apertura, LOW);
}
}

*****
*****
```

