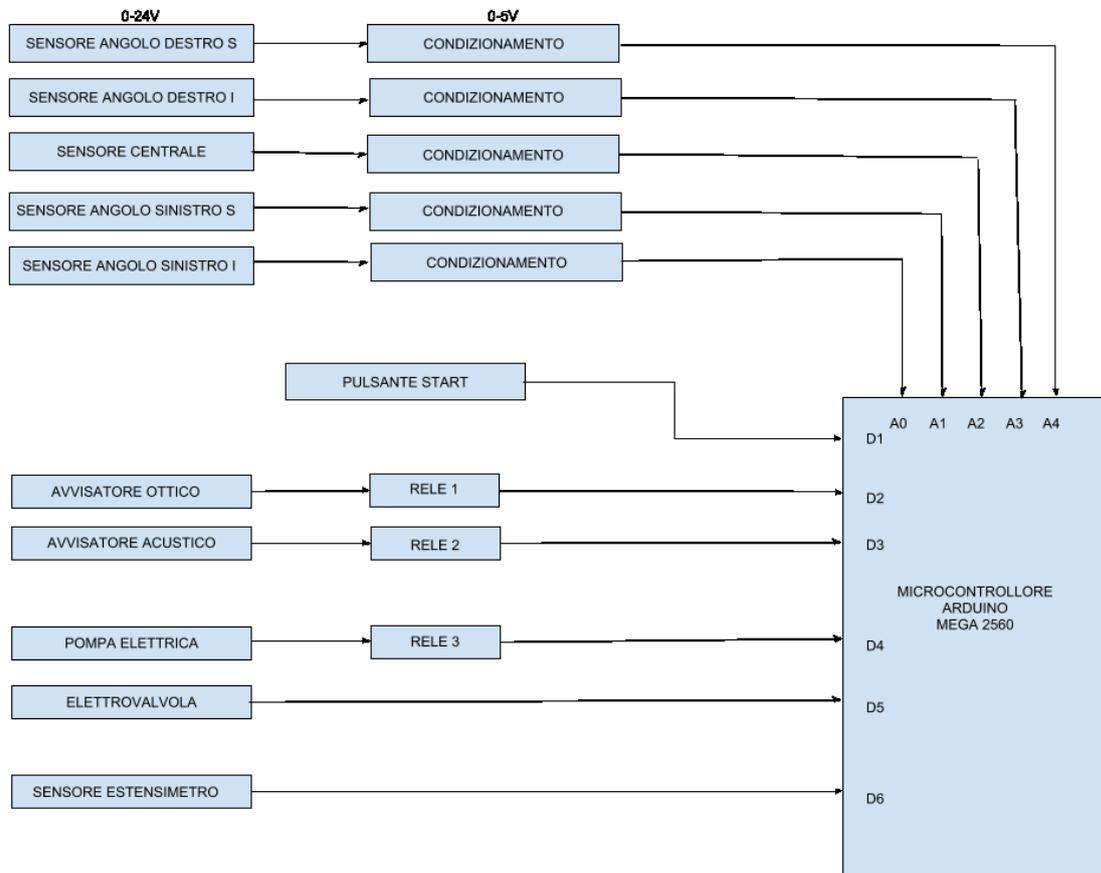


## Schema a blocchi



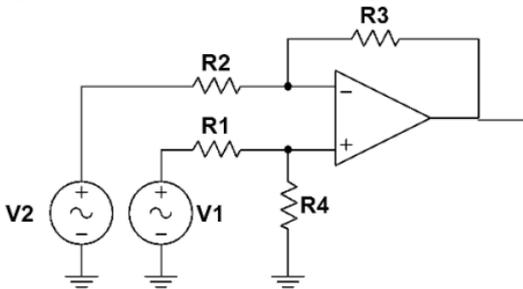
### PUNTO 2)

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 sensore 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:

$$V_{out} = ( V1 - V2 ) * R3/R2$$

La nostra differenza massima  $V1 - V2$  vale 0-24 V che dobbiamo trasformarla in 0-5 V per l'ingresso al microcontrollore , abbiamo che :  $5 = 24 * R3/R2 \rightarrow R3/R2 = 5/24$  , per cui , teoricamente  $R3$  dovrebbe valere 5 k $\Omega$  e  $R2 = 24$  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)

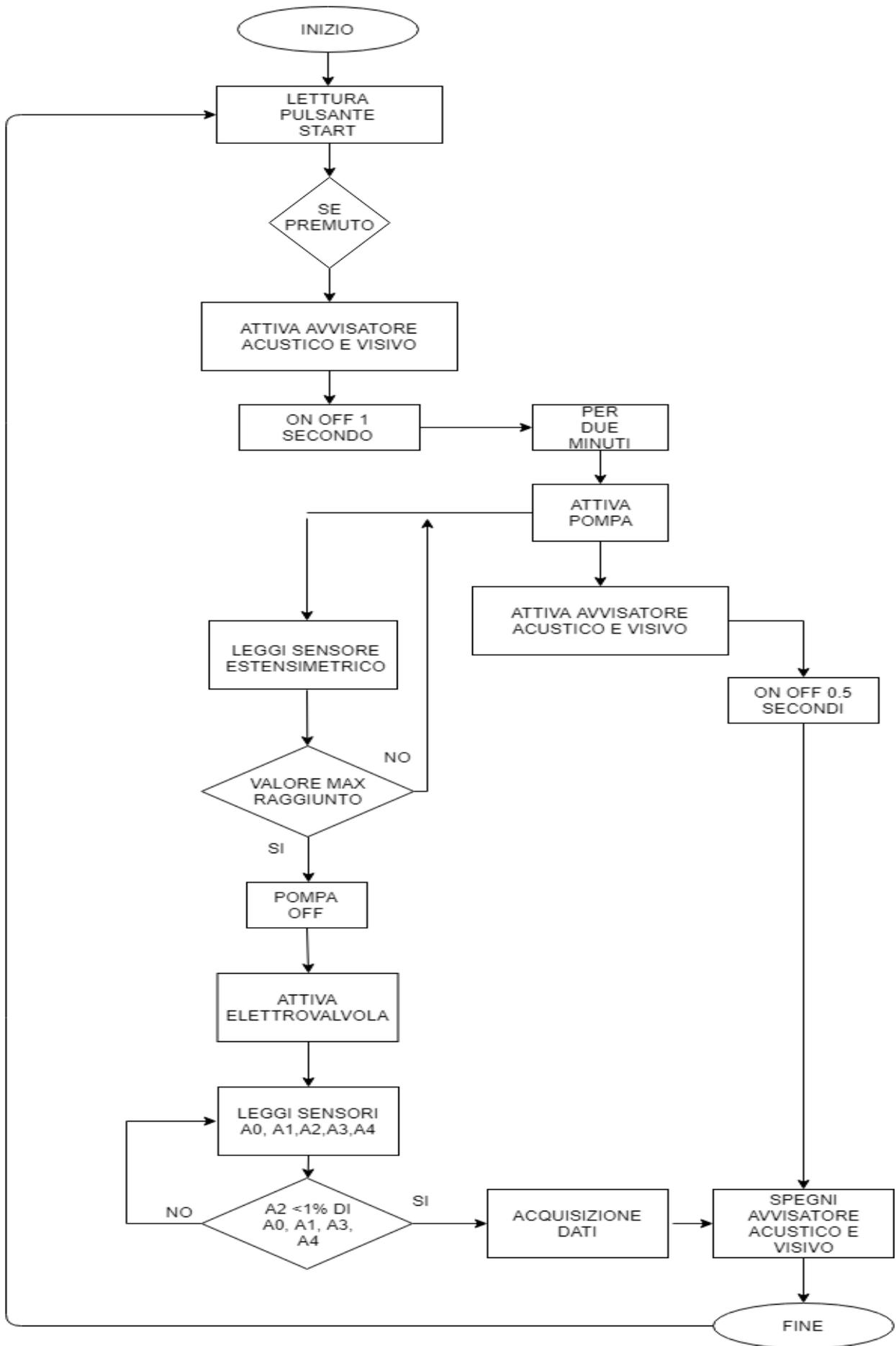
Per il tipo di microcontrollore si assume Arduino La funzione millis() restituisce il numero di millisecondi da quando la scheda Arduino ha iniziato l'esecuzione del programma TIMEOUT è una costante e vale 120'000 ms in quanto corrisponde a 120 secondi dopo il quale deve essere azionata la pompa.

Lo stato del valore della deformazione viene letto dal pin analogico A0. 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.

## Diagramma di flusso (Flow Chart)



# Programma di gestione dell' algoritmo

```
#define puls start 1 //pulsante start al pin digitale 1
#define avv ottico 2 //rele avvisatore ottico al pin2
#define avv acustico 3 //rele avvisatore acustico al pin3
#define pompa 4 //rele pompa al pin 4
#define elettrovalvola //al pin 5
#define sens estensimetro //al pin 6
#define sens A0
#define sens A1
#define sens A2
#define sens A3
#define sens A4
```

```
//Funzioni dei vari cicli
void start();
```

```
//Variabili globali
```

```
int sensorValue = 0;
int reading = 0;
int sensorPin = A0;
int sensorPin = A1;
int sensorPin = A2;
int sensorPin = A3;
int sensorPin = A4;
```

```
//inizio ciclo
```

```
void setup() {
  pinMode(puls start, INPUT); //
  pinMode(avv ottico, OUTPUT); //
  pinMode(avv acustico, OUTPUT); //
  pinMode(pompa, OUTPUT); //
  pinMode(elettrovalvola, OUTPUT); //
  pinMode(sens estensimetro, INPUT); //
```

```
  pinMode(sens A0, INPUT);
  pinMode(sens A1, INPUT);
  pinMode(sens A2, INPUT);
  pinMode(sens A3, INPUT);
  pinMode(sens A4, INPUT);
```

```
//2)
```

```
  serial.println //scrivi sul monitor seriale i valori ricevuti
  delay (60000); //ogni 60 secondi
}
```

```
//Funzioni
```

```

void start(){
{
if(digitalRead(puls start)==HIGH){ // se si preme il pulsante start
digitalWrite (avv ottico,HIGH); //accendi l'avvisatore ottico
digitalWrite (avv acustico,HIGH); //accendi l'avvisatore acustico
delay (1000); //aspetta un secondo
digitalWrite (avv ottico,LOW); //spegni l'avvisatore ottico
digitalWrite (avv acustico,LOW); //spegni l'avvisatore ottico
delay (1000); //aspetta un secondo
}
delay (120000); //ciclo void start per 2 minuti
}

digitalWrite (pompa, HIGH); attiva il pistone
digitalWrite (avv ottico,HIGH); //accendi l'avvisatore ottico
digitalWrite (avv acustico,HIGH); //accendi l'avvisatore acustico
delay (500); //aspetta mezzosecondo
digitalWrite (avv ottico,LOW); //spegni l'avvisatore ottico
digitalWrite (avv acustico,LOW); //spegni l'avvisatore ottico
delay (500); //aspetta mezzo secondo
digitalRead (sens estensimetro); //leggi il sensore estensimetrico

if (sens estensimetrico == valore prestabilito); //se il sensore rileva deformazione superiore al
valore prestabilito:
{
digitalWrite (pompa, LOW);// DISATTIVA il pistone
digitalWrite (elettrovalvola, HIGH);// ATTIVA L'ELETTROVALVOLA e rilascia il fluido
}

if(analogRead(sens A0,A1,A3,A4) >1*100 > Sens A2)
{
//scrivi il valore e acquisisci i dati
serial.println (Valore);
//disattiva segnalazioni acustiche e visive
digitalWrite (avv ottico,LOW); //spegni l'avvisatore ottico
digitalWrite (avv acustico,LOW); //spegni l'avvisatore ottico
}
}
//fine ciclo prova conclusa

```

## Seconda parte:

Inserire 2 quesiti tra quelli proposti nella seconda parte

### Quesito 1:

In ambienti di grandi dimensioni il segnale può avere piccoli ritardi derivati dalla lunghezza del filo conduttore. Nel caso specifico è opportuno che il segnale viaggi su un **conduttore isolato e schermato** fino al dispositivo di **condizionamento del segnale**. Quest'ultimo deve trovarsi nelle **immediate vicinanze del dispositivo di acquisizione** (nel nostro caso Arduino mega) per evitare interferenze e perdite di segnale.

### Quesito 2:

La generazione dei segnali ottici e acustici di preallarme deve essere eseguita da apparecchiature da esterno, dobbiamo ricordare che l'acqua e la polvere possono causare gravi danni.

Per questo sarà necessario utilizzare i materiali adeguati come scatole di derivazione e **portafrutti a tenuta stagna e attrezzi elettrici a doppio isolamento**.

Nel nostro caso possiamo utilizzare:

-per i segnali ottici **lampadine a LED** ad alta efficienza luminosa

-per i segnali acustici una **sirena da esterno**

entrambi devono avere certificazione **IP 68** e devono essere resistenti agli urti.

### Quesito 3:

Con l'acronimo **RAEE** si identificano i rifiuti di apparecchiature elettriche ed elettroniche o semplicemente rifiuti elettronici.

sono rifiuti di tipo particolare che consistono in qualunque apparecchiatura elettrica o elettronica di cui il possessore intenda disfarsi in quanto guasta, inutilizzata, o obsoleta e dunque destinata all'abbandono.

Appartengono in questa categoria:

1. Grandi elettrodomestici;
2. Piccoli elettrodomestici;
3. Apparecchiature informatiche e per telecomunicazioni;
4. Apparecchiature di consumo;
5. Apparecchiature di illuminazione;
6. Strumenti elettrici ed elettronici (ad eccezione delle macchine utensili industriali fisse di grandi dimensioni);
7. Giocattoli e apparecchiature per lo sport e per il tempo libero;
8. Dispositivi medici (ad eccezione di tutti i prodotti impiantati ed infetti);
9. Strumenti di monitoraggio e controllo;
10. Distributori automatici.

Il trattamento dei RAEE è svolto in centri adeguatamente attrezzati, autorizzati alla gestione dei rifiuti ed adeguati al "Decreto RAEE", sfruttando le migliori tecniche disponibili.

Le attività di trattamento prevedono varie fasi, indicativamente:

- messa in sicurezza o **bonifica**, ovvero asportazione dei componenti pericolosi
- smontaggio dei sotto-assiemi e **separazione** preliminare dei materiali
- lavorazione meccanica per il **recupero** dei materiali.

#### Quesito 4:

Il Piano Operativo Sicurezza, definito anche con l'acronimo di **POS**, è un documento redatto dal **Datore di Lavoro** di un'impresa esecutrice, che contiene le informazioni specifiche di ogni cantiere in cui opera l'azienda, oltre che una valutazione dei rischi a cui sono sottoposti gli addetti.

**Il POS è redatto a cura di ciascun datore di lavoro delle imprese esecutrici**, ai sensi dell'articolo 17 del presente Decreto, e successive modificazioni, in riferimento al singolo cantiere interessato; esso contiene almeno i seguenti elementi:

a) i dati identificativi dell'impresa esecutrice, che comprendono:

- 1) il nominativo del datore di lavoro, gli indirizzi ed i riferimenti telefonici della sede legale e degli uffici di cantiere;
- 2) la specifica attività e le singole lavorazioni svolte in cantiere dall'impresa esecutrice e dai lavoratori autonomi subaffidatari;
- 3) **i nominativi degli addetti al pronto soccorso**, antincendio ed evacuazione dei lavoratori e, comunque, alla gestione delle emergenze in cantiere, del rappresentante dei lavoratori per la sicurezza, aziendale o territoriale, ove eletto o designato;
- 4) il **nominativo del medico competente** ove previsto;
- 5) il nominativo del **responsabile del servizio di prevenzione e protezione; RSPP**
- 6) i nominativi del direttore tecnico di cantiere e del capocantiere;
- 7) il numero e le relative qualifiche dei lavoratori dipendenti dell'impresa esecutrice e dei lavoratori autonomi operanti in cantiere per conto della stessa impresa;

Uso di millis()

```
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) ;
inMode(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); }  
  
}
```