



# **Manuale d'uso**

## **Sistema di gestione e di controllo piscina per fisioterapia riabilitativa**

Version 1.0

Andrea Battista  
14 febbraio 2023

# Indice

<b>1</b>	<b>Generalità</b>	<b>2</b>
<b>2</b>	<b>Componenti e attrezzatura utilizzata</b>	<b>3</b>
2.1	Tabella . . . . .	3
2.2	Arduino Mega 2560 Rev3 . . . . .	4
2.3	TMP 36 . . . . .	5
2.3.1	Gestione TMP36 nel software . . . . .	6
2.3.2	Software TMP36 . . . . .	7
2.4	PH-metro . . . . .	8
2.4.1	Gestione PH-metro nel software . . . . .	9
2.4.2	Software PH-metro . . . . .	9
2.5	Display 16x2 . . . . .	10
2.5.1	Gestione display . . . . .	11
2.5.2	Software display . . . . .	11
2.6	Rele' . . . . .	12
2.6.1	Gestione rele' . . . . .	13
2.6.2	Software rele' . . . . .	13
2.7	Riscaldatore . . . . .	14
2.8	Valvola solenoide . . . . .	15
<b>3</b>	<b>Schema a blocchi</b>	<b>16</b>
<b>4</b>	<b>Schema di collegamento</b>	<b>17</b>
<b>5</b>	<b>Schema di montaggio</b>	<b>18</b>
<b>6</b>	<b>Software</b>	<b>19</b>
6.1	Inizializzazione del sistema . . . . .	20
6.2	Loop function . . . . .	21
6.3	Funzioni misura . . . . .	22
6.4	Funzioni di stampa temperatura . . . . .	23
6.5	Funzioni di stampa PH . . . . .	24
<b>7</b>	<b>Referenze</b>	<b>25</b>
7.1	TMP36 . . . . .	25
7.2	PH-metro . . . . .	25
7.3	Valvola solenoide . . . . .	25
7.4	Riscaldatore . . . . .	25
7.5	Rele' . . . . .	25

# 1 Generalità

Benvenuti nel manuale d'uso della gestione della piscina per la fisioterapia di riabilitazione. Questa guida vi fornirà informazioni dettagliate su come utilizzare al meglio il sistema di controllo della temperatura e del valore del pH della piscina per garantire un'esperienza di terapia confortevole e sicura.

Il sistema è progettato per mantenere una temperatura ottimale per la riabilitazione e vi permetterà di controllare facilmente la temperatura dell'acqua. In caso di temperatura troppo bassa, il sistema accenderà automaticamente un riscaldatore per aumentare la temperatura dell'acqua. Se invece la temperatura è troppo alta, il sistema vi darà la possibilità di aggiungere acqua fredda per raffreddare l'acqua.

Il sistema di gestione della piscina controlla anche il valore del pH dell'acqua, che è un fattore importante per garantire un'esperienza sicura e confortevole. In caso di valore del pH troppo elevato o troppo basso, il sistema vi darà la possibilità di intervenire con l'aggiunta di cloro per mantenere il valore del pH nella gamma ottimale.

In questo manuale troverete istruzioni passo-passo per l'utilizzo del sistema, nonché informazioni sulle funzionalità e sulla manutenzione.

## 2 Componenti e attrezzatura utilizzata

### 2.1 Tabella

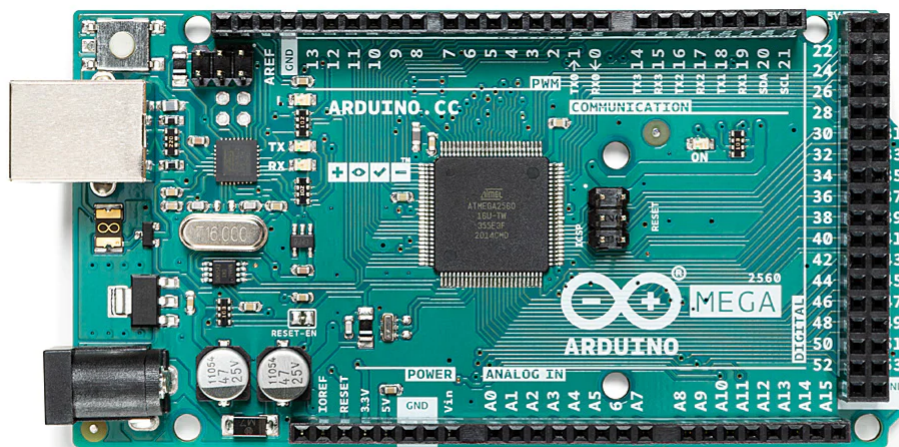
Componenti	Descrizione	Note
Arduino Mega 2560	Scheda microcontrollore	
TMP36	Sensore di temperatura	Utilizzabile nei liquidi
Riscaldatore		Gestito da un relè
Ph-metro	Sensore di rilevazione ph	SEN0161
Valvola solenoide	Motore per azionare pompa	Gestito da un relè
Display 16x2	Display LCD, 2 righe, 16 caratteri	
Resistore		x2, utilizzati per il display
Rele'	Commutatore a comando elettrico.	x2

## 2.2 Arduino Mega 2560 Rev3

L'Arduino Mega 2560 Rev3 è una scheda di sviluppo microcontrollore basata sul microcontrollore ATmega2560. È stata progettata per fornire un'ampia gamma di funzionalità per progetti di automazione e controllo. Con 54 pin di ingresso/uscita digitali, 16 canali di ingresso analogico e una memoria flash di 256 KB, l'Arduino Mega 2560 offre molte opzioni per i progetti che richiedono una quantità significativa di dati e funzionalità.

L'Arduino Mega 2560 è compatibile con una vasta gamma di sensori e dispositivi esterni, rendendolo un'ottima scelta per progetti di automazione e controllo come sistemi di illuminazione, sistemi di sicurezza, robot e molto altro ancora. Inoltre, è dotato di una interfaccia USB che permette di programmare la scheda e di trasferire i dati tra la scheda e il computer.

Uno dei vantaggi dell'Arduino Mega 2560 è la sua semplicità di utilizzo. Grazie alla sua vasta community di sviluppatori, esistono molte risorse online che forniscono guide e tutorial per aiutare i principianti a utilizzare la scheda. Inoltre, la programmazione dell'Arduino Mega 2560 è semplice grazie all'ambiente di sviluppo integrato (IDE) basato su software open source e alla vasta libreria di codici pronti all'uso.

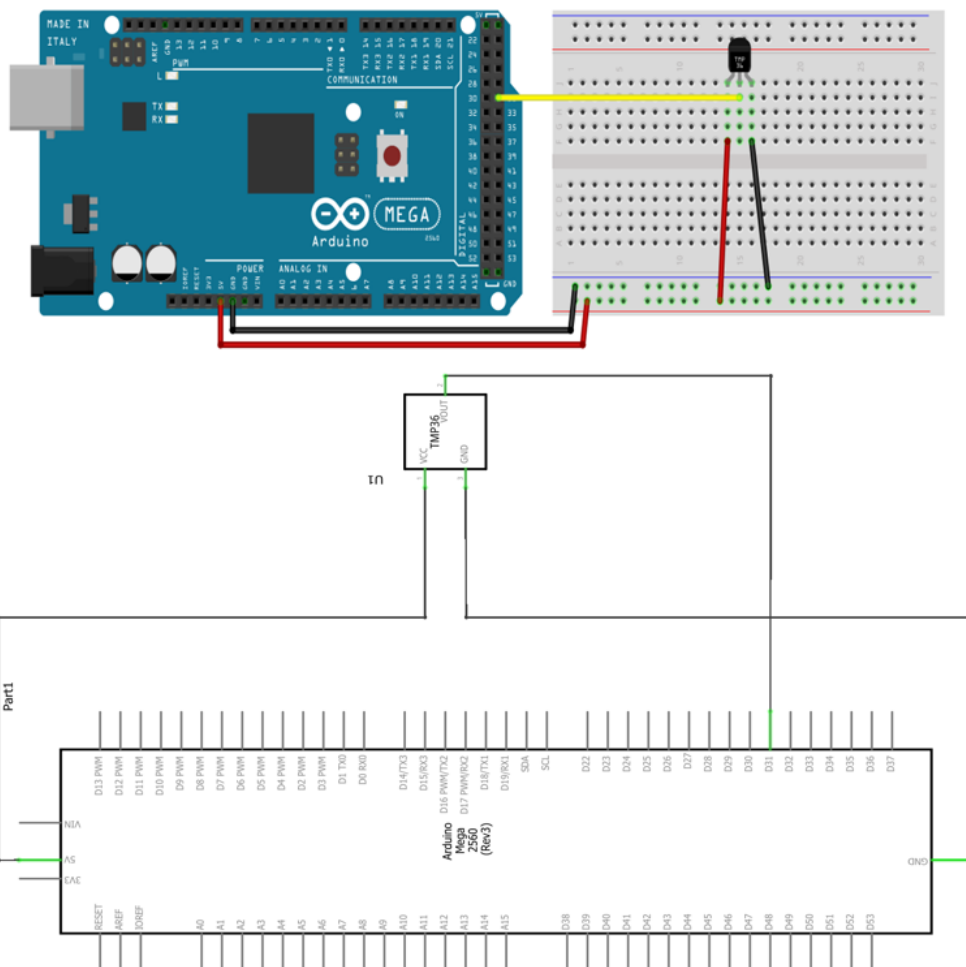


## 2.3 TMP 36

Il TMP36 è un sensore di temperatura a basso costo e a bassa potenza prodotto da Analog Devices. Questo sensore utilizza un circuito integrato che converte la temperatura in un segnale di tensione proporzionale alla temperatura. Questa tensione può essere facilmente letta da un microcontrollore come l'Arduino utilizzando un pin analogico.

Il TMP36 è molto preciso, con una precisione di  $\pm 2^{\circ}\text{C}$ . Inoltre, è progettato per funzionare in una vasta gamma di temperature, da  $-40^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$ , rendendolo adatto a molteplici applicazioni, dalla misura della temperatura ambiente a quella di un motore.

Il TMP36 è anche molto semplice da utilizzare. Per utilizzare il TMP36 con un microcontrollore, basta connettere i pin di alimentazione, massa e segnale al microcontrollore, quindi utilizzare una semplice formula matematica per convertire la tensione in gradi Celsius. Questo rende il TMP36 una scelta ideale per progetti di automazione e controllo che richiedono la misura della temperatura.



### 2.3.1 Gestione TMP36 nel software

Il codice utilizza la libreria Arduino standard per leggere i dati dai sensori di temperatura TMP36.

La prima riga "define SENSNUM 8" definisce una costante chiamata "SENSNUM" con il valore 8. Questo numero rappresenta il numero di sensori TMP36 che verranno utilizzati nel progetto.

Successivamente, sono definite due variabili di tipo intero "i" e "ypin". La variabile "i" viene utilizzata come contatore nel ciclo "for" per leggere i valori dai sensori. La variabile "ypin" rappresenta il numero di pin analogico a cui sarà connesso il primo sensore.

Il codice include anche tre array di tipo float: "valadc", "temp" e "mediaTemp". L'array "valadc" memorizzerà i valori letti dai sensori in formato ADC, mentre l'array "temp" memorizzerà i valori di temperatura convertiti. La variabile "mediaTemp" sarà utilizzata per memorizzare la media dei valori di temperatura rilevati dai sensori.

La funzione "setup" inizializza la comunicazione seriale alla velocità di 9600 baud.

La funzione "loop" richiama la funzione "misuroTemperatura", che esegue la misurazione effettiva della temperatura. La funzione "misuroTemperatura" legge i valori di temperatura dai sensori TMP36 utilizzando un ciclo "for" e il metodo "analogRead". Successivamente, i valori di temperatura vengono convertiti in gradi Celsius utilizzando una formula matematica. Infine, la funzione calcola la media dei valori di temperatura e la restituisce come risultato

### 2.3.2 Software TMP36

```
#define SENSNUM 8    // numero sensori temperatura

float mediaTemp = 0; // temperatura media

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

void loop() {
    misuroTemperatura();
}

double misuroTemperatura(){
    int i = 0;
    int ypin = 54;
    float val_adc[SENSNUM], temp[SENSNUM]; // array temperature

    for(i=0;i<SENSNUM;i++){
        val_adc[i] = analogRead(ypin); // lettura tensione su ypin
        temp[i] = ((val_adc[i] * 0.00488) - 0.5) / 0.01; // tensione in decimale
        ypin++; // incremento ypin
    }

    for(i=0;i<SENSNUM;i++){
        mediaTemp += temp[i]; // somma di tutte le temperature
    }

    mediaTemp = mediaTemp / SENSNUM; // media temperature
    return mediaTemp;
}
```

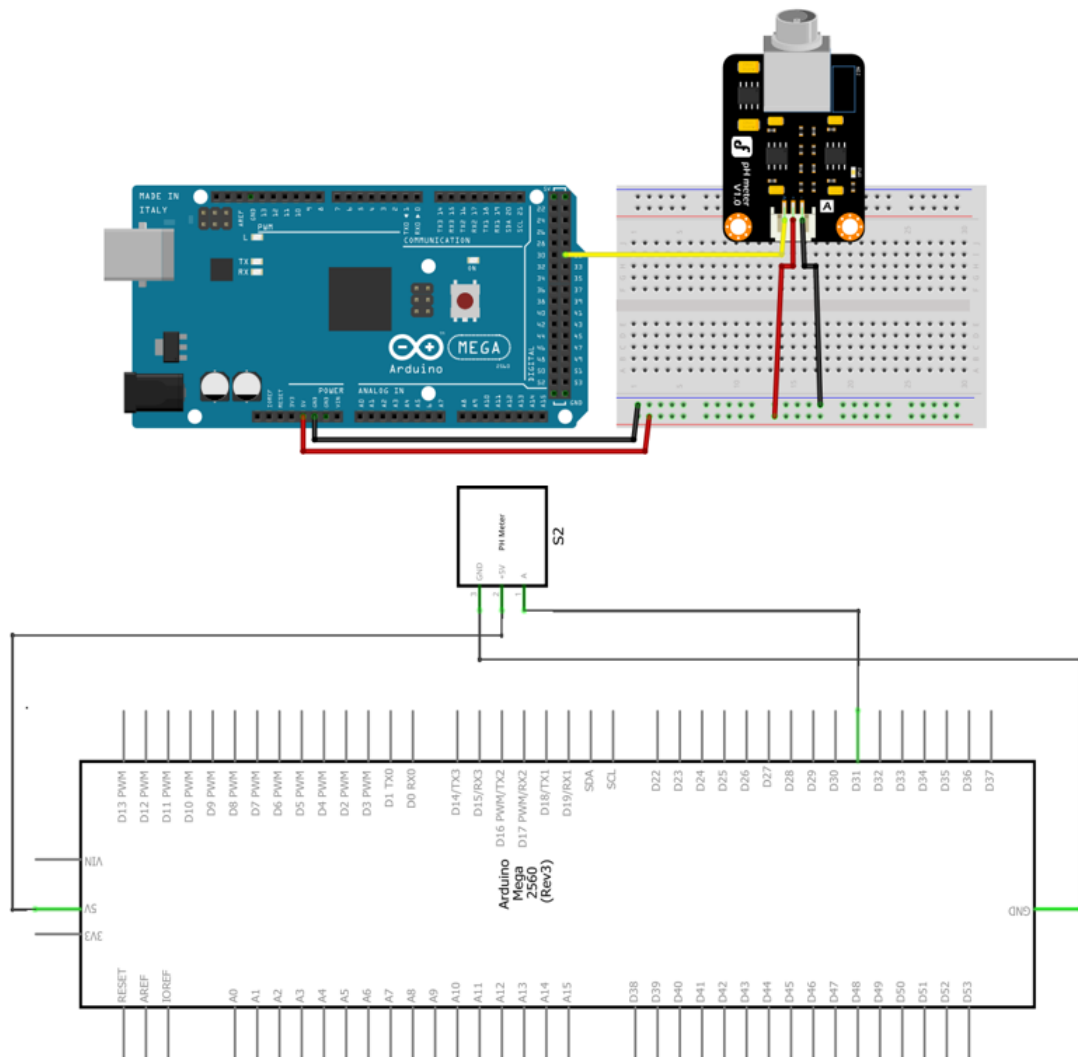


## 2.4 PH-metro

Un pH-metro è un dispositivo che viene utilizzato per misurare il pH di una soluzione, ovvero la concentrazione di ioni di idrogeno presenti in essa. Il pH è una scala che va da 0 a 14, dove un valore di 7 rappresenta una soluzione neutra, mentre valori inferiori indicano una soluzione acida e valori superiori indicano una soluzione basica.

Ci sono diverse tipologie di pH-metri, ma la maggior parte utilizza un sensore che misura la differenza di potenziale tra un elettrodo di riferimento e un elettrodo di misura immersi nella soluzione. Questi elettrodi sono costituiti da materiali che reagiscono con gli ioni di idrogeno, generando un segnale elettrico proporzionale alla concentrazione di ioni di idrogeno. Questo segnale viene quindi elaborato dal pH-metro, che visualizza il risultato come valore di pH.

Per questo progetto e' stato utilizzato il SEN0161



### 2.4.1 Gestione PH-metro nel software

Il sensore di pH è collegato alla porta digitale 7 dell'Arduino e la modalità di questa porta viene impostata come "OUTPUT" nel blocco setup().

La funzione valorePh viene utilizzata per leggere il valore dal sensore di pH utilizzando la funzione analogRead(PHSENS).

Il valore letto viene quindi convertito in volt utilizzando la formula  $\text{voltage} = \text{phValue} * (5.0 / 1023.0)$ . Infine, il valore di pH viene calcolato utilizzando la formula  $\text{ph} = 3.5 * \text{voltage} + 7.0$ .

### 2.4.2 Software PH-metro

```
#define PHSENS 7    // pin ph-metro

// variabili globali
float ph = 0;      // valore ph

void setup() {
  pinMode(PHSENS, OUTPUT); // inizializzazione ph-metro
}

void loop() {
  misuroPh();
}

double misuroPh(){
  float phValue, voltage = 0;

  phValue = analogRead(PHSENS); // lettura tensione sensore pH
  voltage = phValue * (5.0 / 1023.0); // conversione del valore in volt
  ph = 3.5 * voltage + 7.0; // valore di pH in formato decimale

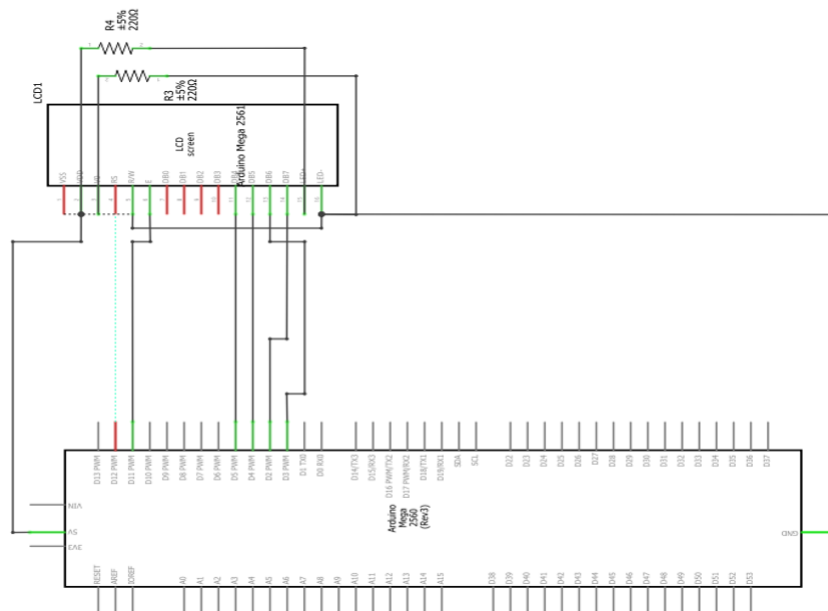
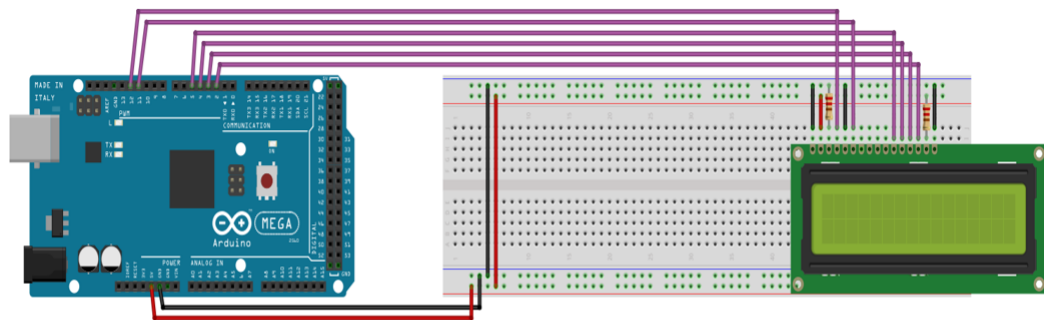
  return ph;
}
```

## 2.5 Display 16x2

Un display a 16x2 è un display LCD che ha 16 colonne e 2 righe di caratteri. Può essere utilizzato con un microcontrollore come l'Arduino per mostrare informazioni come temperature, livelli di pH, conteggi e molto altro.

Per utilizzare un display a 16x2 con Arduino, è necessario collegare i pin del display al microcontrollore utilizzando una libreria specifica per la gestione del display. Questa libreria fornisce funzioni per scrivere sullo schermo, modificare il cursore e altre funzionalità.

Una volta configurato il display e la libreria, è possibile scrivere sul display utilizzando codice personalizzato, ad esempio, per visualizzare le informazioni sulla temperatura o il livello di pH.



### 2.5.1 Gestione display

Questo codice include la libreria LiquidCrystal e crea un oggetto LCD che viene inizializzato nella funzione setup().

La funzione print viene utilizzata per stampare il testo "Hello" sulla prima riga del display LCD. La funzione setCursor viene utilizzata per impostare la posizione del cursore sulla seconda riga, e la funzione print viene chiamata di nuovo per stampare il testo "World" sulla seconda riga.

La funzione loop() è vuota, il che significa che una volta eseguito il setup, il testo verrà stampato una volta e non verrà più aggiornato.

IL display e' stato utilizzato per stampare molte informazioni, per questo se si e' interessati a tutte le stampe per il progetto, ai punti 6.4 e 6.5 del manuale troverete tutte le funzioni di stampa.

### 2.5.2 Software display

```
#include <LiquidCrystal.h>

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

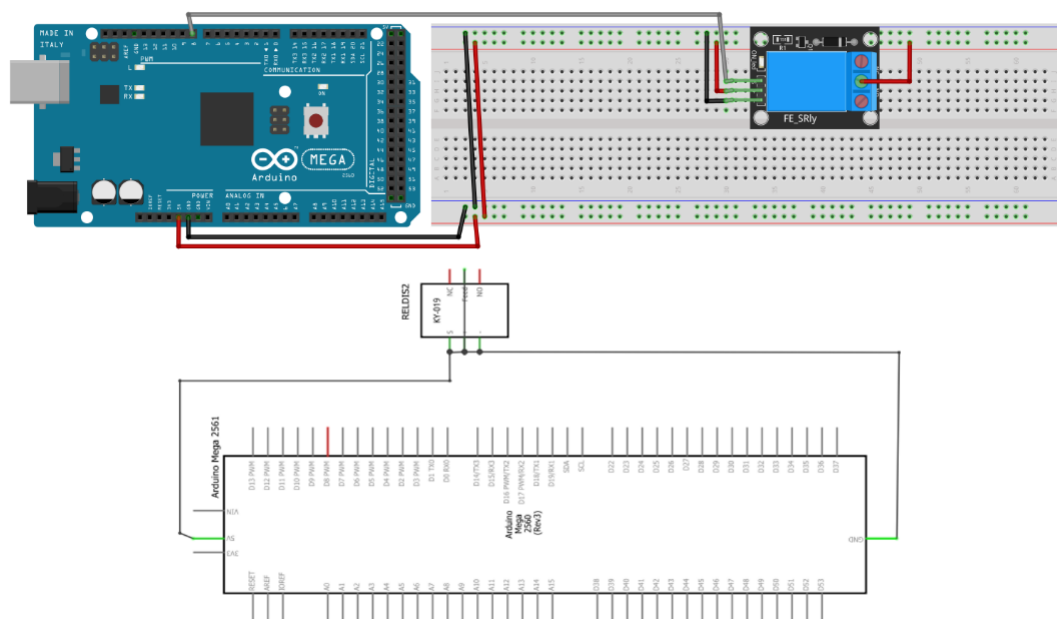
void setup() {
  lcd.begin(16, 2);
  lcd.print("Hello");
  lcd.setCursor(0, 1);
  lcd.print("World");
}

void loop() {}
```

## 2.6 Relè'

Un relè è un componente elettronico che permette di controllare l'alimentazione di un carico, come una lampadina, un motore o un altro dispositivo, tramite un segnale elettrico di controllo. Il relè è composto da un'armatura magnetica che si muove quando viene attivato da un segnale elettrico, interrompendo o completando un circuito.

Per utilizzare un relè con un Arduino, è necessario collegare il polo positivo del relè al pin di alimentazione (es. 5V), il polo negativo al pin di GND e il segnale di controllo al pin di output del microcontrollore.



### 2.6.1 Gestione rele'

La prima riga di codice definisce una costante intera chiamata "relayPin" e la imposta a 8. Questo significa che la costante "relayPin" sempre valrà 8 e non potrà essere modificata.

La funzione "setup" viene eseguita una sola volta quando il dispositivo viene acceso o reset. In questa funzione, la funzione "pinMode" viene chiamata per impostare il pin 8 come un pin di output.

La funzione "loop" viene eseguita continuamente una volta che la funzione "setup" è stata completata. In questa funzione, la funzione "digitalWrite" viene chiamata per impostare il pin 8 su un valore logico alto (HIGH) per un secondo. Quindi, viene impostato il pin su un valore logico basso (LOW) per un altro secondo. Questo crea un ciclo che alterna continuamente tra i valori alti e bassi ogni secondo.

### 2.6.2 Software rele'

```
const int relayPin = 8;

void setup() {
  pinMode(relayPin, OUTPUT);
}

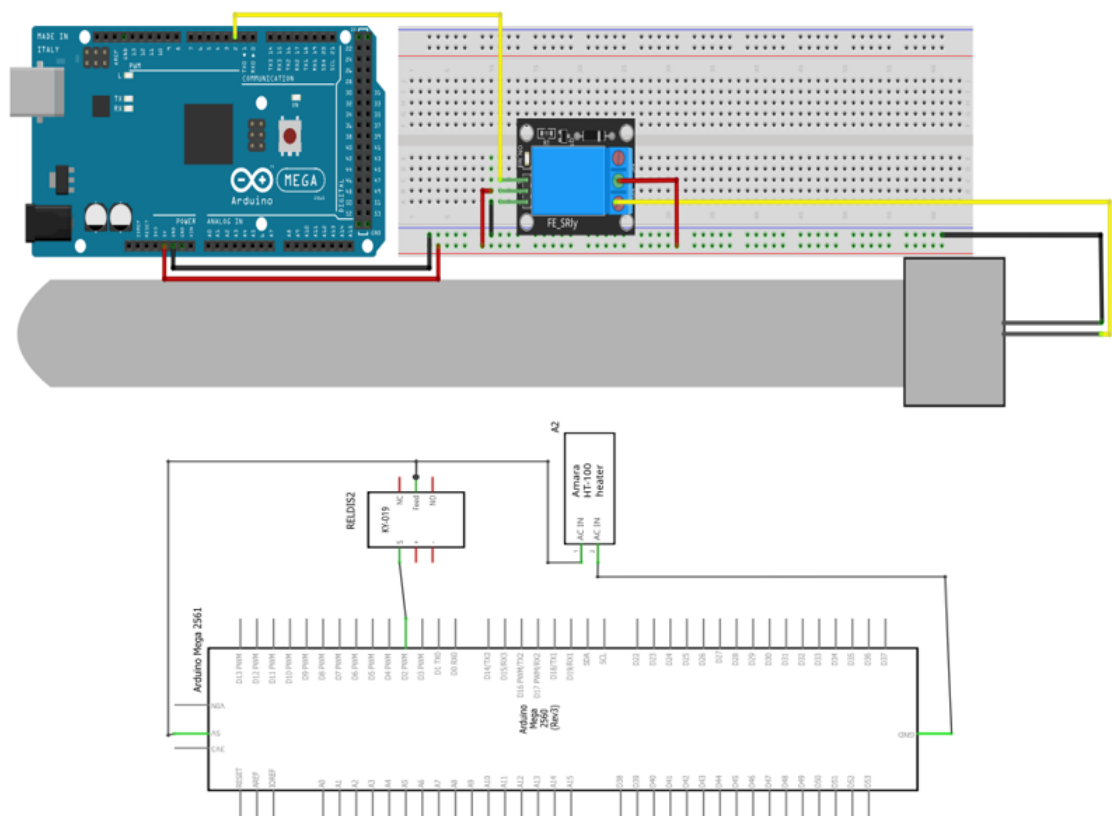
void loop() {
  digitalWrite(relayPin, HIGH);
  delay(1000);
  digitalWrite(relayPin, LOW);
  delay(1000);
}
```

## 2.7 Riscaldatore

Un riscaldatore per piscina è un dispositivo che viene utilizzato per riscaldare l'acqua della piscina al fine di renderla più confortevole per il nuoto o per altre attività in acqua. Ci sono diversi tipi di riscaldatori per piscina, ma i più comuni sono quelli a gas, elettrici e a energia solare.

I riscaldatori elettrici utilizzano l'elettricità per riscaldare l'acqua della piscina. Questi sono solitamente più economici da acquistare e da utilizzare rispetto ai riscaldatori a gas, ma sono anche meno potenti e richiedono più tempo per riscaldare l'acqua della piscina.

La gestione del rel' avviene tramite un rel', che provvede ad attivare o disattivare il dosatore in base all'input che riceve da Arduino. Per vedere la gestione del rel' si puo' andare nella sezione [Rele'](#)

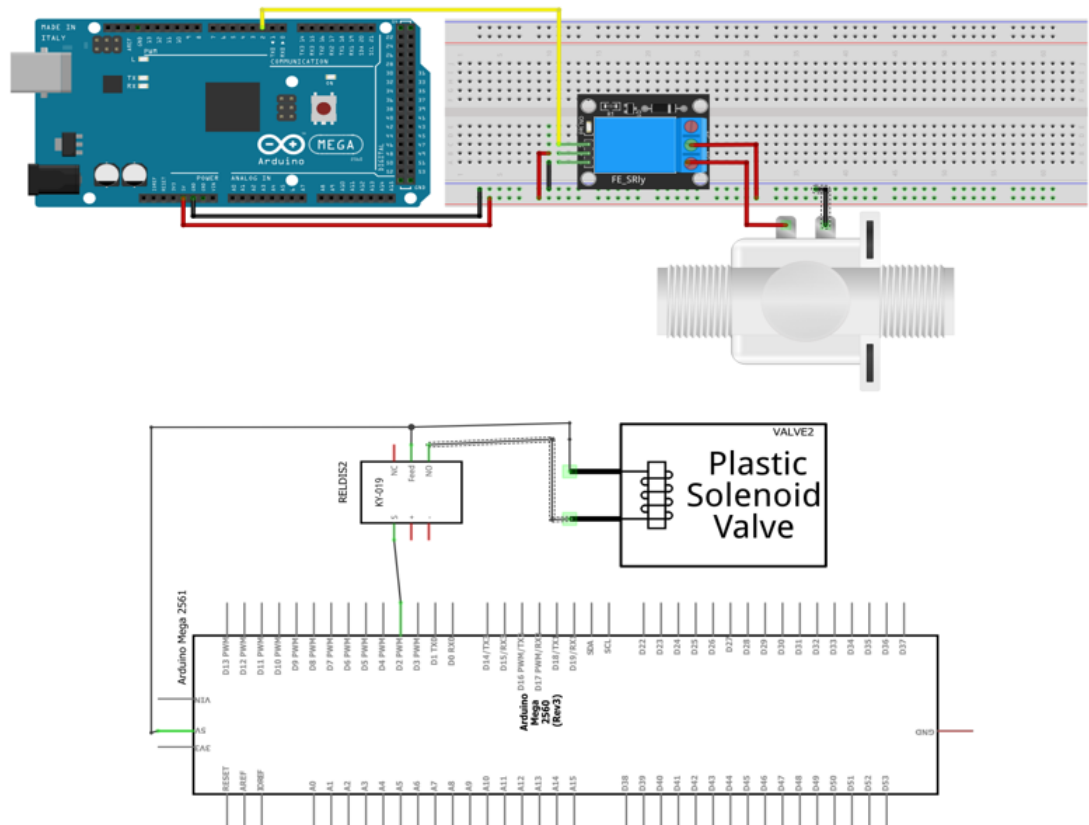


## 2.8 Valvola solenoide

Una valvola solenoide è un dispositivo elettromeccanico che utilizza un campo magnetico generato da una corrente elettrica per aprire o chiudere un flusso di liquido o gas in un sistema controllato.

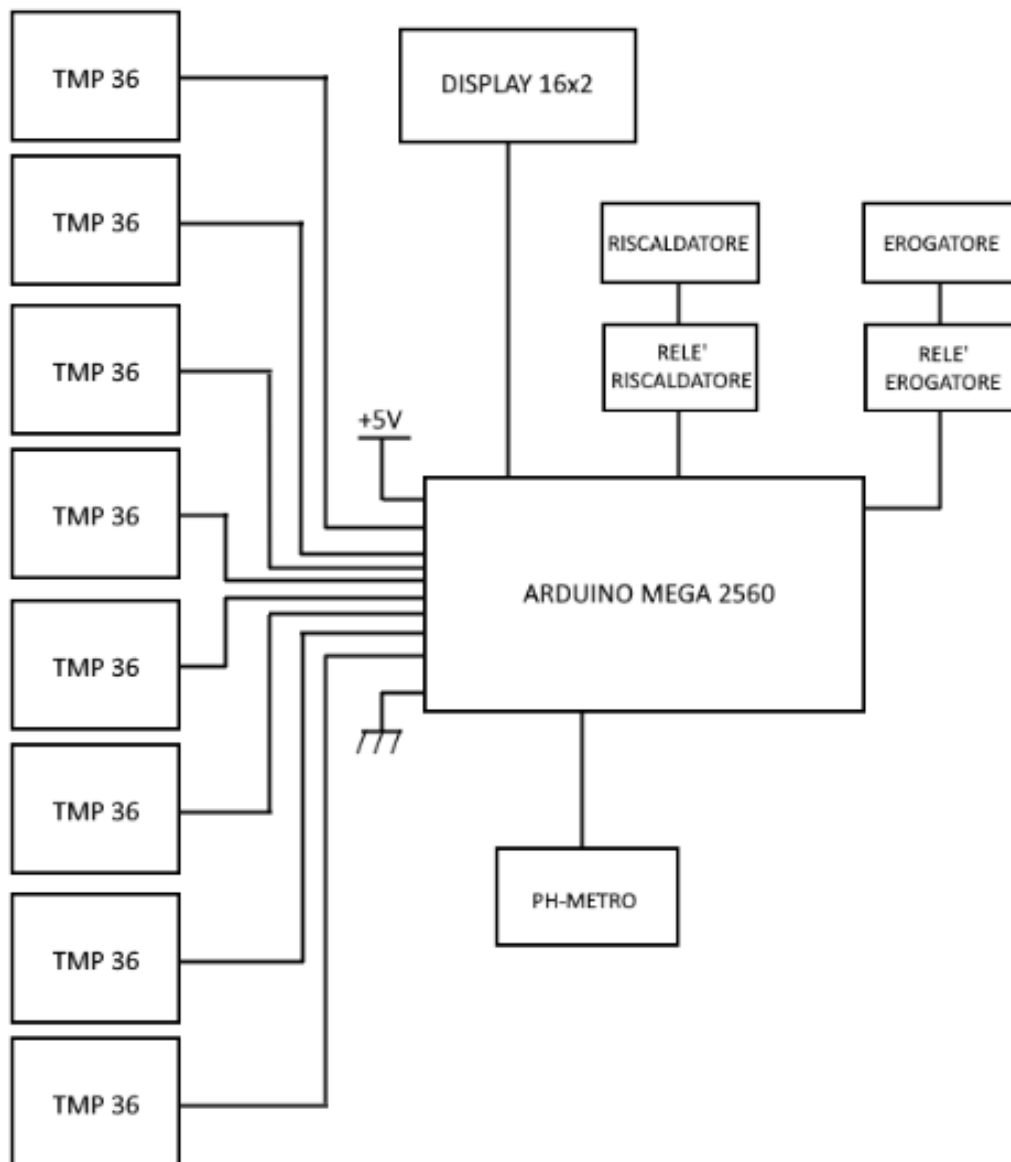
Per utilizzare una valvola solenoide con Arduino bisogna collegare la valvola solenoide all'Arduino. La valvola solenoide richiede un'uscita di corrente sufficiente per attivare l'elettromagnete, quindi sarà necessario utilizzare un transistor o un relè per pilotare la corrente. Per vedere la gestione del relè si può andare nella sezione [Relè](#)

È importante notare che la configurazione esatta del circuito e del codice dipenderà dal tipo di valvola solenoide utilizzata e dal sistema che si desidera controllare. È sempre consigliabile consultare il datasheet del componente e verificare che il circuito sia sicuro e funzionante prima di utilizzarlo.

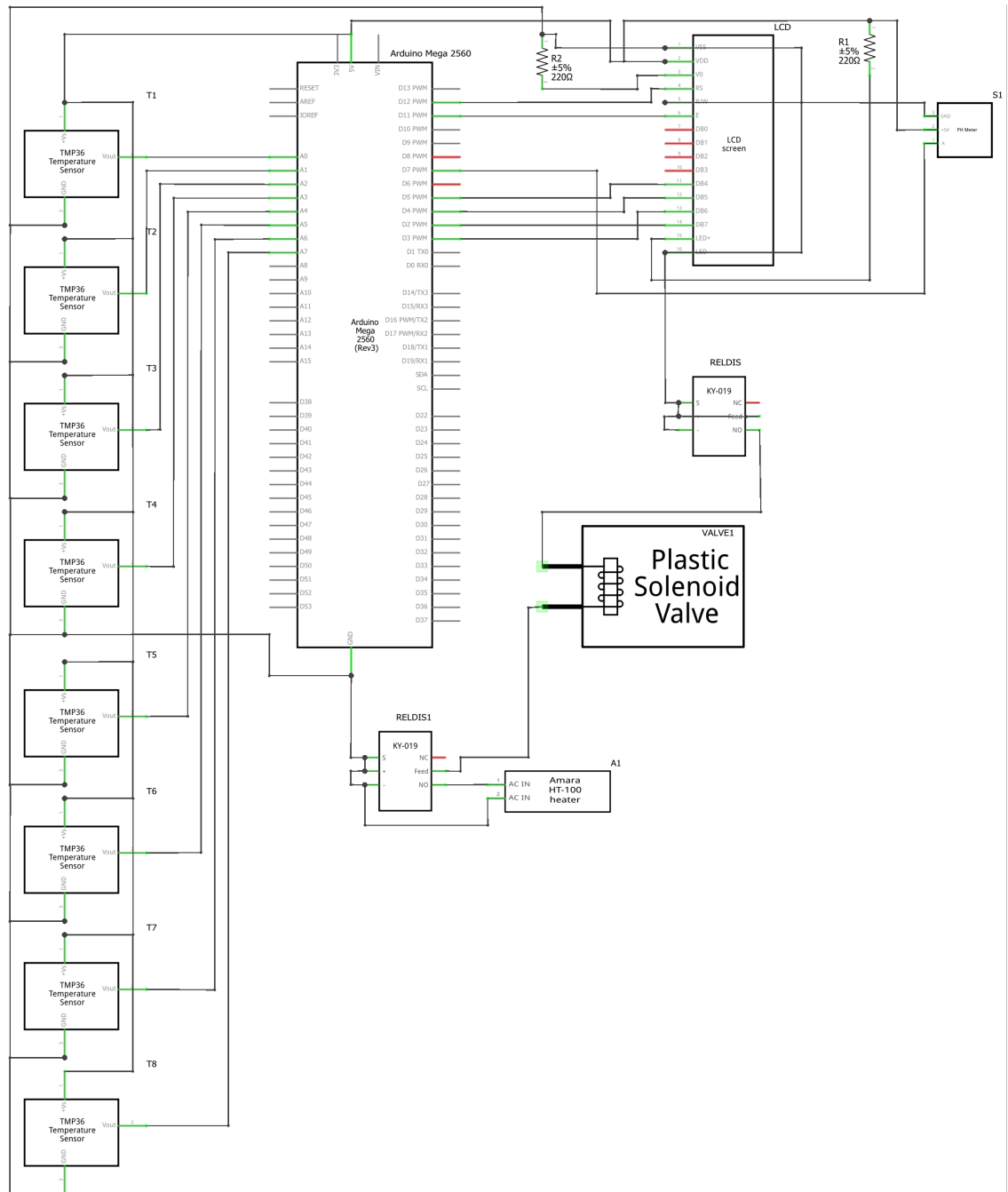




### 3 Schema a blocchi

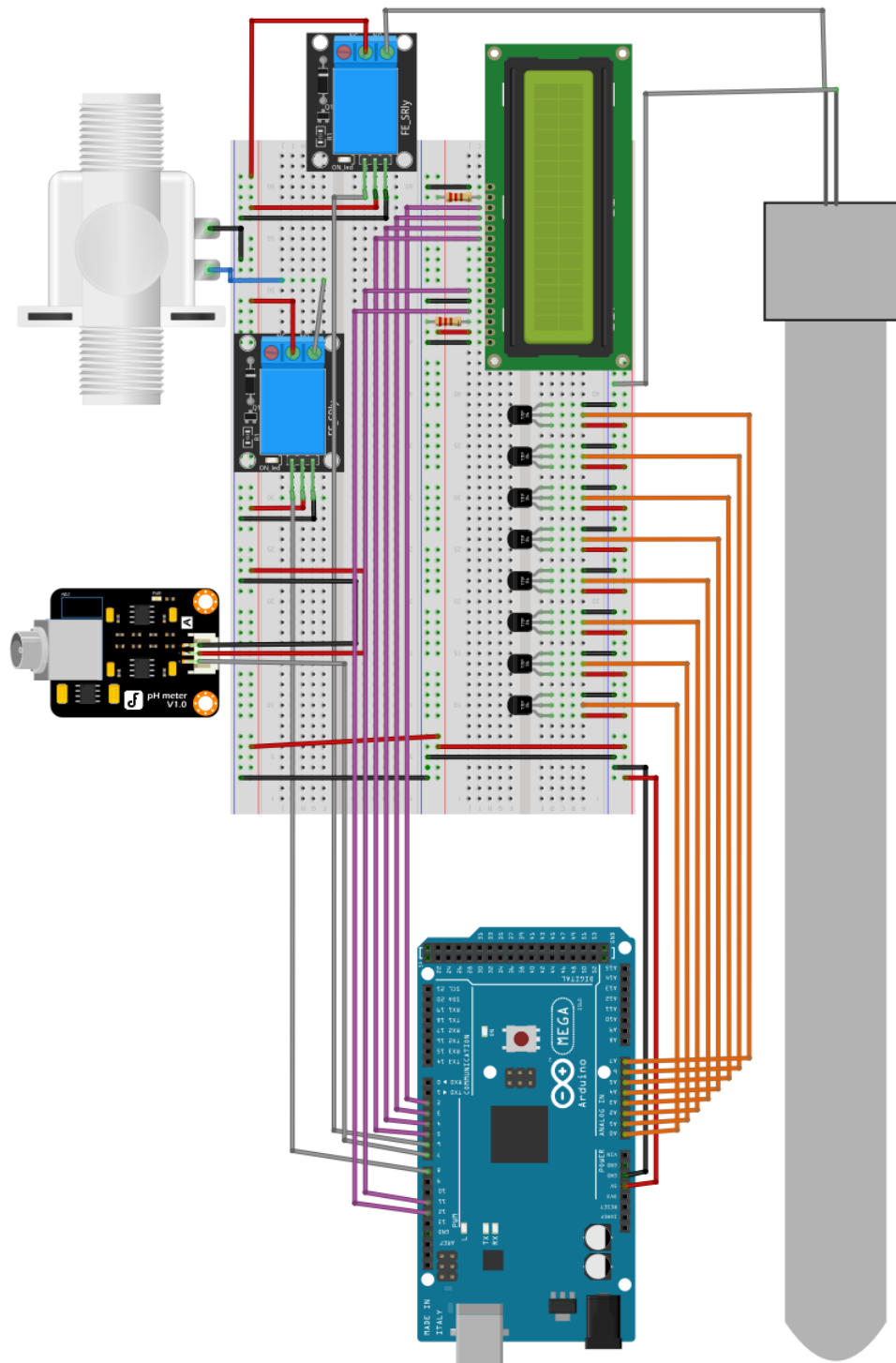


## 4 Schema di collegamento



fritzing

## 5 Schema di montaggio



## 6 Software

Questo codice è un programma per Arduino che gestisce il controllo della temperatura e del pH di un acquario. Inizialmente, la temperatura viene misurata utilizzando un sensore TMP36 e viene visualizzata su un display LCD.

Se la temperatura è troppo bassa, il riscaldatore viene acceso e la temperatura viene rilevata di nuovo.

Se la temperatura è troppo alta, il riscaldatore viene spento e viene visualizzato un messaggio di errore.

Dopo che la temperatura è stata controllata, il pH viene misurato utilizzando un sensore pH e viene visualizzato su un display LCD.

Se il pH è troppo alto, viene attivata una valvola solenoide per abbassare il pH.

Il codice include anche la libreria LiquidCrystal, che fornisce le funzioni per l'utilizzo del display LCD. Inoltre, vengono definite alcune costanti, come le tolleranze per temperatura e pH, e i pin sull'Arduino per i sensori e gli attuatori. Durante la funzione di impostazione, vengono inizializzati il monitor seriale, il display LCD e i pin per i sensori e gli attuatori. La funzione principale "loop" viene eseguita continuamente e contiene le istruzioni per il controllo della temperatura e del pH. Ci sono anche alcune funzioni supplementari, come "misuroTemperatura", "stampoTemperatura" e "phAlto", che aiutano a gestire la misurazione e la visualizzazione dei dati.

## 6.1 Inizializzazione del sistema

```
#include <LiquidCrystal.h>

#define SENSNUM 8    // numero sensori temperatura
#define TMIN 26      // temperatura minima
#define TMAX 28      // temperatura massima
#define PHSENS 7     // pin ph-metro
#define PHMIN 6.8    // ph minimo
#define PHMAX 7.4    // ph massimo
#define RELRISC 6    // pin rele riscaldatore
#define RELDOS 8     // pin rele dosatore

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);    // pin display

// variabili globali
float mediaTemp = 0;    // temperatura media
float ph = 0;           // valore ph

void setup() {
    Serial.begin(9600);    // inizializzazione monitor seriale
    pinMode(PHSENS, OUTPUT);    // inizializzazione ph-metro
    pinMode(RELRISC, OUTPUT);    // inizializzazione riscaldatore
    pinMode(RELDOS, OUTPUT);    // inizializzazione dosatore
}
```

## 6.2 Loop function

```
void loop() {
  misuroTemperatura();
  stampoTemperatura();
  delay(2000);

  if(mediaTemp < TMIN){
    do{
      temperaturaBassa();           // stampa temperatura bassa
      digitalWrite(RELRISC, LOW); // accensione riscaldatore
      delay(5000);
      misuroTemperatura();
      stampoTemperatura();
    }while(mediaTemp < TMIN);
  }
  else if(mediaTemp > TMAX){
    temperaturaAlta();           // stampa temperatura alta
    digitalWrite(RELRISC, LOW); // spegnimento riscaldatore
  }
  else{}

  misuroPh();
  stampoPh();
  delay(2000);

  if(ph > PHMAX){
    do{
      phAlto();           // stampa ph alto
      digitalWrite(RELDOS, HIGH); // accensione dosatore
      misuroPh();
      stampoPh();
      delay(500);
    }while(ph > PHMAX);

    digitalWrite(RELDOS, LOW); // spegnimento dosatore
    misuroPh();
    stampoPh();
    delay(500);
  }
}
```

## 6.3 Funzioni misura

```
double misuroTemperatura(){
    int i = 0;
    int ypin = 54;
    float val_adc[SENSNUM], temp[SENSNUM]; // array temperature

    for(i=0;i<SENSNUM;i++){
        val_adc[i] = analogRead(ypin); // lettura tensione su ypin
        temp[i] = ((val_adc[i] * 0.00488) - 0.5) / 0.01; // tensione in decimale
        ypin++; // incremento ypin
    }

    for(i=0;i<SENSNUM;i++){
        mediaTemp += temp[i]; // somma di tutte le temperature
    }

    mediaTemp = mediaTemp / SENSNUM; // media temperature
    return mediaTemp;
}

double misuroPh(){
    float phValue, voltage = 0;

    phValue = analogRead(PHSENS); // lettura tensione sensore pH
    voltage = phValue * (5.0 / 1023.0); // conversione del valore in volt
    ph = 3.5 * voltage + 7.0; // valore di pH in formato decimale

    return ph;
}
```

## 6.4 Funzioni di stampa temperatura

```
// stampa valore medio temperatura
void stampoTemperatura(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp acqua:\t");
    lcd.setCursor(0, 1);
    lcd.print(mediaTemp);
    lcd.print(" ^C");
}

// stampa errore temperatura bassa
void temperaturaBassa(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("T acqua bassa:");
    lcd.setCursor(0, 1);
    lcd.print("Accendo riscald");
}

// stampa errore temperatura alta
void temperaturaAlta(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("T acqua elevata:");
    lcd.setCursor(0, 1);
    lcd.print("Spengo riscald");

    delay(2000);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Spengo riscald");
    lcd.setCursor(0, 1);
    lcd.print("Aggiungere acqua");
}
```



## 6.5 Funzioni di stampa PH

```
void stampoPh(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Valore PH:\t");
    lcd.setCursor(0, 1);
    lcd.print(ph);
}

void phAlto(){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Valore PH alto");
    lcd.setCursor(0, 1);
    lcd.print("Accendo dosatore");
}
```

## **7 Referenze**

### **7.1 TMP36**

<https://logicaprogrammabile.it/arduino-sensore-di-temperatura-tmp36/>

### **7.2 PH-metro**

<https://www.intrageo.it/geologia/ph-dellacqua-come-misurarlo-con-arduino/>  
<https://forum.fritzing.org/t/probe-ph-sensor/14575/2>

### **7.3 Valvola solenoide**

<https://logicaprogrammabile.it/arduino-sensore-di-temperatura-tmp36/>

### **7.4 Riscaldatore**

<https://forum.fritzing.org/t/water-heater-aquarium/16035/8>

### **7.5 Rele'**

<https://www.antoniovasco.it/2020/03/07/il-rele/#:~:text=Un%20rel%20un%20componente,su%20un%20circuito%20a%20220>