

IIS ARCHIMEDE



MANUALE D'USO

SISTEMA DI RILEVAMENTO TEMPERATURA E
UMIDITA' TRAMITE SENSORI

MATTIA BOSI 5^N A.S. 2022/2023

INDICE

<u>1.GENERALITA</u>	<u>Pag.3</u>
<u>2.SCHEMA A BLOCCHI</u>	<u>Pag.4</u>
<u>3.LISTA COMPONENTI</u>	<u>Pag.5</u>
<u>DHT11</u>	<u>Pag.6</u>
<u>AD590</u>	<u>Pag.6</u>
<u>RISCALDATORE</u>	<u>Pag.6</u>
<u>VENTOLA</u>	<u>Pag.6</u>
<u>DISPLAY LCD</u>	<u>Pag.6</u>
<u>4.SCHEMA ELETTRICO</u>	<u>Pag.7</u>
<u>5.SOFTWARE</u>	<u>Pag.8-14</u>
<u>6.BIBLIOGRAFIA</u>	<u>Pag.15</u>

1.GENERALITA'

Il sistema realizzato permette di rilevare valori di temperatura e umidità tramite appositi sensori. Ha due modalità, Automatico e Manuale che viene scelta tramite un interruttore ON-OFF.

Il valore di temperatura viene prelevato tramite al sensore analogico AD590 ed al sensore digitale DHT11.

Il valore di umidità viene prelevato anch'esso tramite il sensore DHT11.

I valori di temperatura e umidità dovranno essere memorizzati in due vettori di 30 elementi.

Per il valore di temperatura, essendo prelevato da due sensori diversi, prima di essere salvato nel vettore, viene fatta una media dei due valori prelevati, e nel vettore ci sarà la media dei due valori prelevati

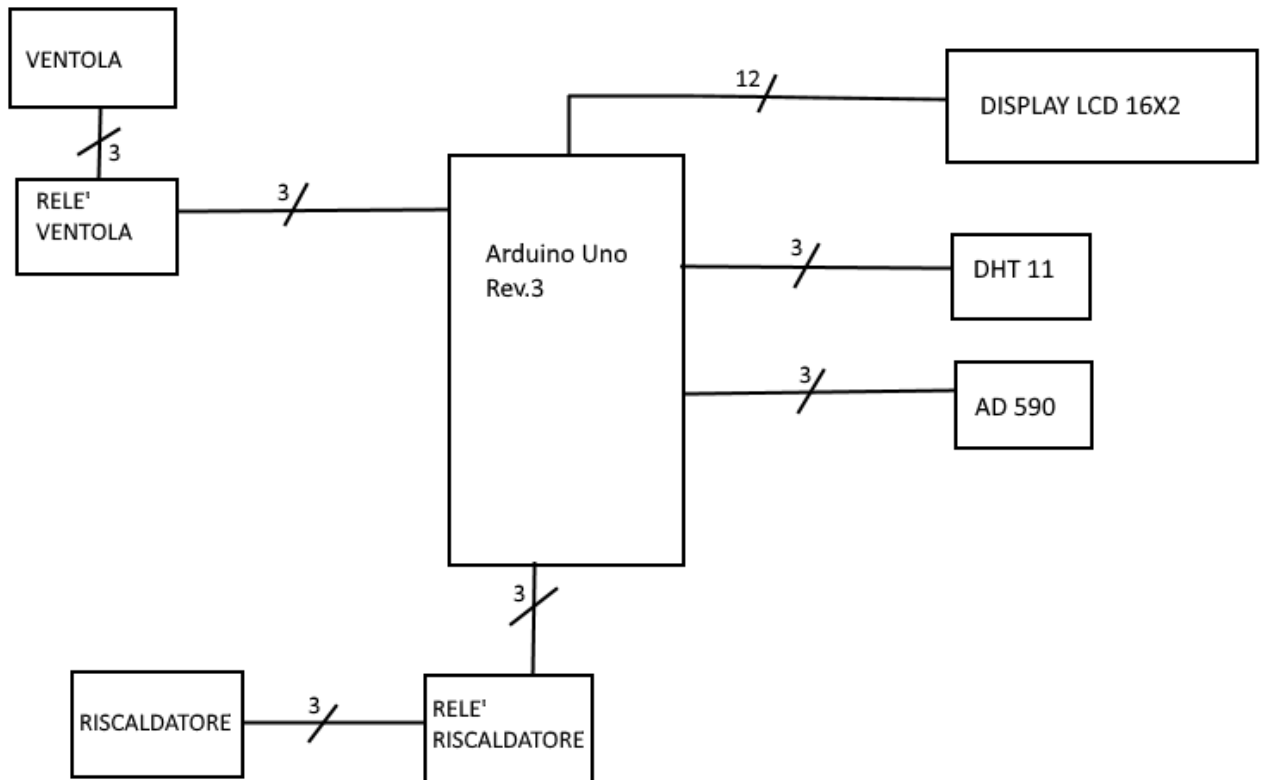
Infine, verrà fatta una media di tutti i valori presenti all'interno del vettore sia per la temperatura sia per l'umidità che verrà stampata su un display LCD dedicato.

In modalità automatica il valore medio della temperatura deve essere compreso tra 19° e 24°. Se la temperatura scende sotto i 19° attivare un riscaldatore collegato ad un relè. Il riscaldatore deve essere simulato da un led rosso;

Se la temperatura supera i 24° attivare una ventola collegata ad un relè. La ventola deve essere simulata da un led verde;

In modalità manuale i valori vengono impostati tramite 2 interruttori dedicati. Se l'interruttore si trova a livello LOW, la temperatura massima o minima(in base a quale interruttore utilizzo) inizierà un conteggio partendo da 0. Quando sono arrivato al valore desiderato mi basta impostare a livello HIGH l'interruttore e il valore verrà salvato nella variabile.

2.SCHEMA A BLOCCHI



3.LISTA COMPONENTI

COMPONENTE	PART NUMBER	NOTE				
SENSORE DHT11		Sensore digitale per la rilevazione della temperatura				
SENSORE AD 590		Sensore analogico per la rilevazione della temperatura				
Relè x2	KY 019	da 5V DC a max 250V AC 10A				
Ventola Automatica		si attiva quando la temperatura è alta				
Riscaldatore Automatico		si attiva quando la temperatura è bassa				
Display LCD		Per la visualizzazione dei valori				
Resistenze Display LCD						
<table><tr><td>R</td><td>220Ω ± 5 % ¼W</td></tr><tr><td>PotR</td><td>10kΩ ± 5 % ¼W</td></tr></table>	R	220Ω ± 5 % ¼W	PotR	10kΩ ± 5 % ¼W		
R	220Ω ± 5 % ¼W					
PotR	10kΩ ± 5 % ¼W					
Resistenze Interruttori						
<table><tr><td>R(interruttore auto/ manuale)</td><td>10kΩ ± 5 % ¼W</td></tr><tr><td>Rx2 per modifica valori temp</td><td>10kΩ ± 5 % ¼W</td></tr></table>	R(interruttore auto/ manuale)	10kΩ ± 5 % ¼W	Rx2 per modifica valori temp	10kΩ ± 5 % ¼W		
R(interruttore auto/ manuale)	10kΩ ± 5 % ¼W					
Rx2 per modifica valori temp	10kΩ ± 5 % ¼W					

SEGUE UNA BREVE DESCRIZIONE DI OGNI SINGOLO COMPONENTE

SENSORE DHT11

Il sensore DHT 11 è un sensore di temperatura e umidità digitale che permette il prelevamento di valori di temperatura e umidità tramite i pin analogici di Arduino

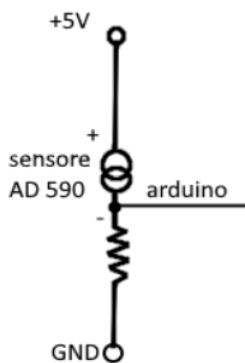
Per il sensore DHT11 esiste una libreria dedicata utilizzabile scrivendo `#include <DHT.h>`

Il vantaggio del sensore DHT11 è che ci fornirà subito un valore in temperatura o umidità, ma lo svantaggio è che non è un sensore molto preciso siccome possiede una Precisione di temperatura: $\pm 2^{\circ}\text{C}$ ed una Precisione di umidità: $\pm 5\%$

SENSORE AD590

Il sensore AD590 è un sensore di temperatura analogico che permette il prelevamento di valori di temperatura tramite i pin analogici di Arduino.

Fornisce un'uscita in corrente quindi per visualizzare i valori in gradi centigradi occorre mettere una resistenza collegata a massa tra il pin negativo del sensore e il cavetto collegato ad Arduino



Dopo aver fatto questo occorre mappare il valore prelevato. Mappare un valore significa convertire il valore analogico in tensione in un valore numerico. Segue un esempio della sintassi della funzione `map()`

```
int temperatura = map(analogRead(A0), 0, 1023, 0, 100);
```

0 e 1023 è l'intervallo di valori di input che ci restituisce l'`analogRead`, 0 e 100 è l'intervallo dei valori di temperatura.

1 RISCALDATORE AUTOMATICO CONTROLLATO CON UN RELE'

Il riscaldatore automatico ha il compito di intervenire quando la temperatura è più bassa del valore minimo (19°C in modalità automatica).

Si è scelto di utilizzare un Relè per comandare il riscaldatore in quanto è facilmente gestibile con Arduino. Non si è potuto collegare direttamente il riscaldatore ad Arduino in quanto funziona con la tensione di rete (220V);

1 VENTOLA AUTOMATICA CONTROLLATA CON UN RELE'

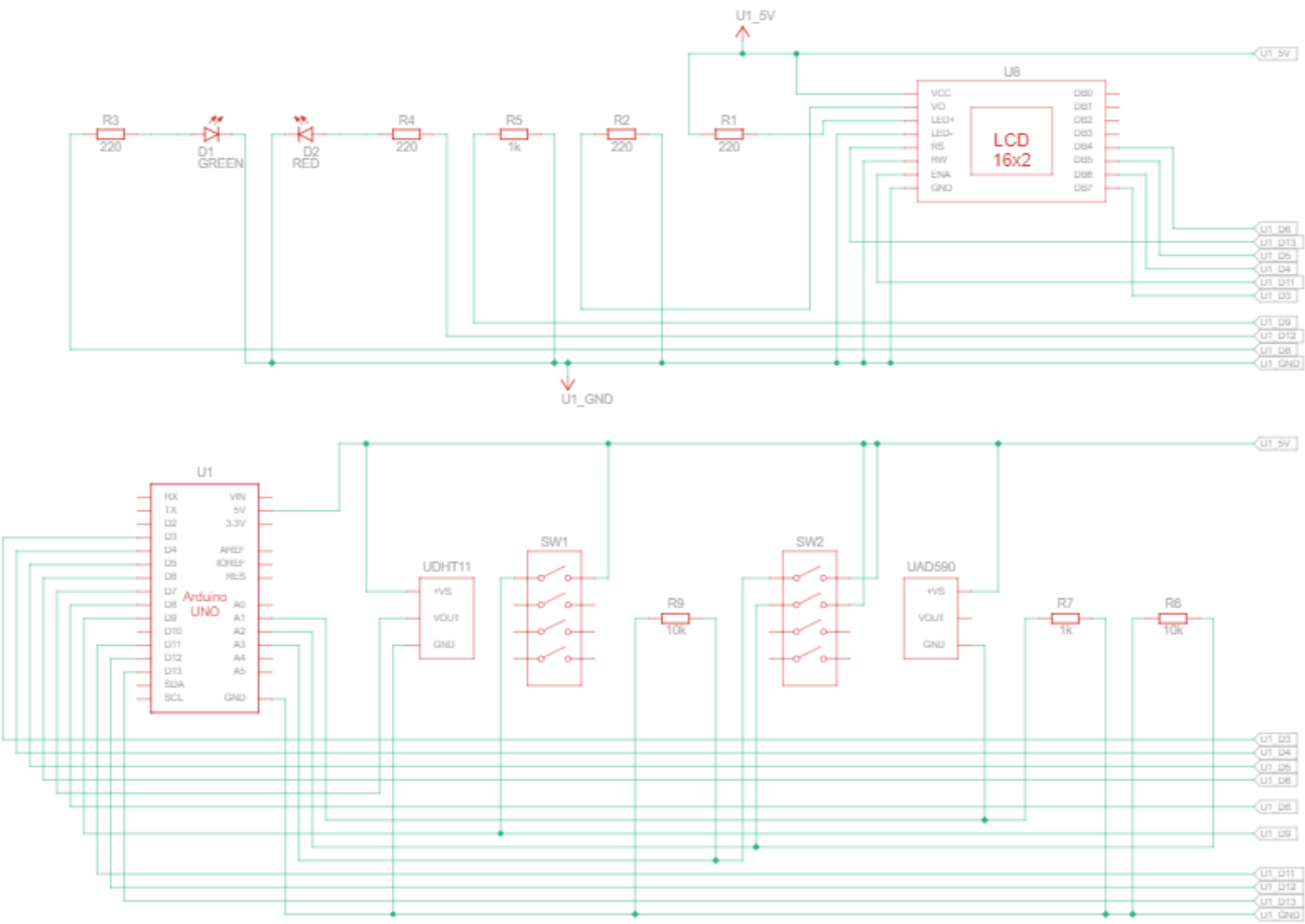
La ventola ha il compito di intervenire quando la temperatura è più alta del valore massimo (24°C in modalità automatica).

Si è scelto di utilizzare un Relè per comandare il riscaldatore in quanto è facilmente gestibile con Arduino. Non si è potuto collegare direttamente il riscaldatore ad Arduino in quanto funziona con la tensione di rete (220V).

DISPLAY LCD

Il display LCD (Liquid Crystal Display) in questo sistema viene utilizzato per mostrare tutti i valori riguardanti la temperatura, il valore dell'umidità e anche che cosa sta succedendo nel sistema in quel momento; quindi, per esempio se il riscaldatore automatico è attivo, se la ventola è attiva ecc....

4.SCHEMA ELETTRICO



5.SOFTWARE

```

/*****
* Seconda Esperienza Secondo Periodo Sistemi Automatici
* Bosi Mattia 5°N
* LA SCHEDA SCELTA PER QUESTO PROGETTO E' ARDUINO UNO
*****/
/*****
* PIN ANALOGICI CHE SI POSSONO UTILIZZARE COME DIGITALI
* A0 = 14 - A1 = 15 - A2 = 16 - A3 = 17 - A4= 18 - A5 = 19
*
*****/
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include "DHT.h"

#include <LiquidCrystal.h>

#define DHTTYPE DHT11
#define DHTPIN 7
#define RELEVENTOLA 8
#define RELERISCALDATORE 12
#define AD590PIN A1
#define DIM 5
#define INTERRUETTORE 9
#define PULSUP 16
#define PULSDOWN 17

int TEMPMAX = 0;
int TEMPMIN = 0;
int modTempMax;
int modTempMin;
float temperatura[DIM];
float umidita[DIM];
float valoreAnalogico[DIM];
float tempIstDht;
float tempIstAd590;
float tempAd590;
float tempAd590Volt;
float valoreUmidita = 0;
float media = 0;
int val2 = 0;
int dim = 5;
int val;
int i;

LiquidCrystal lcd(13,11,6,5,4,3);
DHT dht(DHTPIN,DHTTYPE);

void setup()
{
    Serial.begin(9600);

    //dichiarazione pin

    pinMode(RELEVENTOLA, INPUT);
    pinMode(RELERISCALDATORE, INPUT);
    //pinMode(DHTPIN, INPUT);
    pinMode(AD590PIN, INPUT);
    pinMode(INTERRUPTORE, INPUT);

```



```

pinMode(PULSUP, INPUT);
pinMode(PULSDOWN, INPUT);
TEMPMAX = 24;
TEMPMIN = 19;
modTempMax=0;
modTempMin=0;

//INIZIALIZZAZIONE DISPLAY LCD
lcd.begin(16,2);
lcd.setCursor(0,0);

}

void loop()
{

    int val = digitalRead(INTERRUTTORE); //LETTURA STATO INTERRUTTORE

    delay(1000);

    if(val == 0) //MODALITA AUTOMATICA
    {

        digitalWrite(RELERISCALDATORE, LOW);
        digitalWrite(RELEVENTOLA, LOW);
        lcd.print("MODALITA' ");
        lcd.setCursor(0,1);
        lcd.print("AUTOMATICA");
        Serial.println("MODALITA' AUTOMATICA");

        delay(1000);
        lcd.clear();

        misuroTemperatura();
        misuroUmidita();
        calcoloMedia();

    }

    if(val == 1) //MODALITA MANUALE
    {

        lcd.print("MODALITA' ");
        lcd.setCursor(0,1);
        lcd.print("MANUALE");
        delay(1000);
        lcd.clear();

        modifTemp();

    }

}

float misuroTemperatura() //FUNZIONE CHE ACQUISISCE I VALORI DI TEMPERATURA

```

```

{
i=0;
val == digitalRead(INTERRUETTORE);
int conta = 5;
if(val==0)
{
    while( conta > 0)
    {
        lcd.print("ACQUISIZIONE FRA");
        lcd.setCursor(0,1);

        lcd.print(conta);
        conta--;
        delay(1000);
        lcd.clear();

    }

do
{

    tempAd590Volt = analogRead(AD590PIN);

    tempAd590 = map(tempAd590Volt,0, 1023, 273, 353);
    tempIstAd590 = (tempAd590 - 293);

    tempIstDht = dht.readTemperature();
    //tempIstDht = 22;

    delay(50);
    temperatura[i] = (tempIstDht + tempIstAd590)/2;

    Serial.println(i);

    Serial.print("TEMP DHT = ");
    Serial.print(tempIstDht);
    Serial.println("°C");

    Serial.print("TEMP AD = ");
    Serial.print(tempIstAd590);
    Serial.println("°C");

    lcd.print("MISURAZIONE T");
    lcd.setCursor(0,1);
    lcd.print("IN CORSO...");

    delay(1000);

    i++;
    lcd.clear();

}while(i<DIM);
for(i=0;i<DIM;i++)
{
    Serial.print("TEMP ");
    Serial.print(i);
    Serial.print(" = ");
    Serial.println(temperatura[i]);
    delay(200);
}
}
if(val==1)
{
    lcd.clear();
    delay(1000);
}

```

```

        return 1;
    }

}

float misuroUmidita()
{
    i=0;
    valoreUmidita=0;
    val == digitalRead(INTERRUTTORE);
    if(val==0)
    {
        do
        {
            umidita[i] = dht.readHumidity();

            valoreUmidita = valoreUmidita + umidita[i];
            Serial.println(i);

            Serial.print("UMIDITA' = ");
            Serial.println(umidita[i]);

            lcd.print("MISURAZIONE UM");
            lcd.setCursor(0,1);
            lcd.print("IN CORSO...");
            delay(1000);
            lcd.clear();

            i++;
        }while(i<DIM);

        valoreUmidita = valoreUmidita/DIM;

        Serial.print("MEDIA UMIDITA' = ");
        Serial.println(valoreUmidita);
        lcd.print("UMIDITA = ");
        lcd.print(valoreUmidita);
        delay(2000);
        lcd.clear();
    }
    if(val==1)
    {
        lcd.clear();
        delay(1000);
        return 1;
    }
}

}

float calcoloMedia()
{
    digitalWrite(RELERISCALDATORE, LOW);
    digitalWrite(RELEVENTOLA, LOW);
    val == digitalRead(INTERRUTTORE);
    media = 0;
    if(val==0)
    {
        for(i=0;i<DIM;i++)
        {

```

```

        media = media + temperatura[i];
        delay(200);
    }

    media = media/DIM;

    lcd.print("MEDIA TEMP = ");
    lcd.print(media);
    Serial.print("MEDIA TEMP = ");
    Serial.print(media);
    Serial.println("°C");

    delay(1000);
    lcd.clear();

    if(media < TEMPMIN )
    {
        tempBassa();
    }
    else if (media > TEMPMAX)
    {
        tempAlta();
    }
    else
    {
        lcd.clear();
        lcd.print("TEMP OK...");

        Serial.println("TEMP OK...");

        delay(2000);
        lcd.clear();
    }

}

if(val ==1)
{
    lcd.clear();
    delay(1000);
    return 1;
}

}

void tempBassa()
{
    lcd.clear();
    lcd.print("ATTIVO");
    lcd.setCursor(0,1);
    lcd.print("RISCALDATORE");

    Serial.println("ATTIVO RISCALDATORE");
    digitalWrite(RELERISCALDATORE, HIGH);

    delay(2000);

    lcd.clear();
}

void tempAlta()
{
    lcd.clear();

```

```

    lcd.print("ATTIVO");
    lcd.setCursor(0,1);
    lcd.print("VENTOLA");

    Serial.println("ATTIVO VENTOLA");
    digitalWrite(RELEVENTOLA, HIGH);
    delay(2000);

    lcd.clear();

}

int modifTemp()          //FUNZIONE CHE MODIFICA LA TEMP IN MODALITA MANUALE
{
    i=0;
    val == digitalRead(INTERRUTTORE);

    delay(500);

    if(digitalRead(INTERRUTTORE) == 0)
    {
        return 1;
    }
    else if(digitalRead(INTERRUTTORE) == 1)
    {
        if(digitalRead(PULSDOWN) == HIGH && digitalRead(PULSUP) == LOW ) //MODIFICA TEMP
MAX
        {
            do
            {
                modTempMax = modTempMax +1;
                delay(1000);
                lcd.print("TEMPMAX = ");
                lcd.print(modTempMax);
                delay(1000);
                lcd.clear();

            }while(digitalRead(PULSUP) == LOW);
            lcd.clear();
        }

        delay(1000);

        if(digitalRead(PULSDOWN) == LOW && digitalRead(PULSUP) == HIGH) //MODIFICA TEMP
MIN
        {
            do
            {
                modTempMin = modTempMin +1;
                delay(1000);
                lcd.print("TEMPMIN = ");
                lcd.print(modTempMin);
                delay(1000);
                lcd.clear();

            }while(digitalRead(PULSDOWN) == LOW);
            lcd.clear();
        }

        delay(1000);

        if (digitalRead(PULSDOWN) == HIGH && digitalRead(PULSUP) == HIGH )
        {
            do
            {

```

```

TEMPMIN = modTempMin;
TEMPMAX = modTempMax;

lcd.clear();
lcd.print("TEMPMIN IMPOSTATA");
lcd.setCursor(0,1);
lcd.print(TEMPMIN);
delay(2000);
lcd.clear();

lcd.print("TEMPMAX IMPOSTATA");
lcd.setCursor(0,1);
lcd.print(TEMPMAX);
delay(2000);
lcd.clear();

}while(digitalRead(PULSDOWN) == HIGH && digitalRead(PULSUP) == HIGH &&
digitalRead(INTERRUTTORE) == 1 );

    lcd.clear();
}
}
}

```

6.BIBLIOGRAFIA

[DISPLAY LCD](#)

[DATASHEET AD590](#)

[DHT11](#)