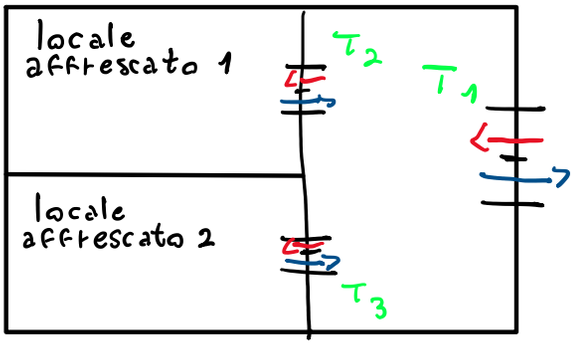
ITAT 2018 - Francesco Barbetta - 5AEC

mercoledì 12 marzo 2025 09:03

È possibile rappresentare il sito archeologico, date le informazioni presenti nella traccia, secondo il seguente schema:



Al fine di controllare gli accessi sono stati utilizzate delle coppie di tornelli forniti di fotocellule. Al passaggio di un visitatore dal tornello viene inviato un impulso a 5V su un pin specifico di un ARDUINO UNO in caso di ingresso, mentre su un altro pin specifico in caso di uscita

(1) Ogni coppia di tornelli è fornita di un proprio ARDUINO UNO che, al raggiungimento del limite massimo di persone, aziona un relè, che permette a una porta automatizzata di bloccarsi dall'esterno, impedendo l'accesso ad altre persone. Ogni tornello ha perciò il medesimo circuito di controllo con relativo codice di programma, con l'unica differenza rappresentata da una variabile che definisce il numero massimo di persone (30 per T_1, 5 per T_2 e T_3).

(2) Il sistema di acquisizione dati è rappresentato dal seguente schema a blocchi:

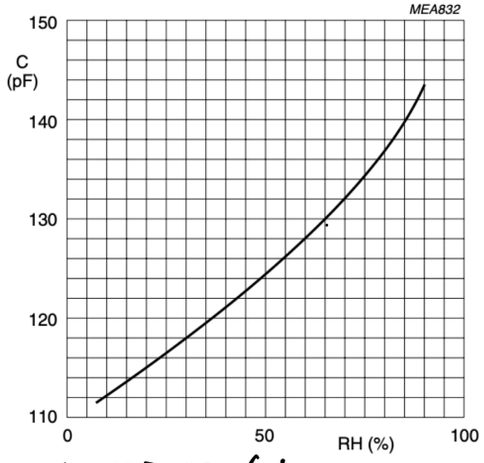
grandezza fisica → [trasduttore] → [circ. di condizionam.] → [pin analog. ARDUINO] → segnale di controllo

Il trasduttore ha il ruolo di convertire la grandezza fisica in una grandezza elettrica. Il circuito di condizionamento serve a convertire la grandezza elettrica, qualora non lo sia già, in una tensione, e, quando utile ad amplificare la tensione stessa in modo da riempire la scala dell'ADC integrato in ARDUINO, col fine ultimo di massimizzare la risoluzione.

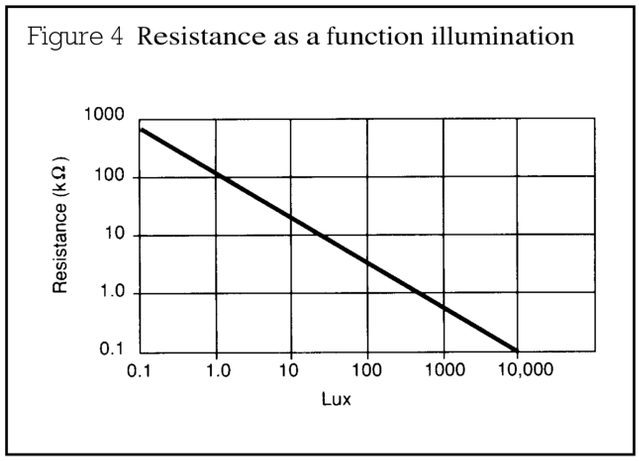
In uno schema generico possono essere presenti multiplexaggio e campionamento. In questo caso il multiplexaggio è inutile data la sufficienza di PIN analogici di ARDUINO UNO. Il campionamento può essere omesso data la variazione lenta delle grandezze fisiche in questione (temperatura, umidità e luminosità).

I trasduttori utilizzati sono:

- TMP 36 per la temperatura → 500 mV a 0°C + 10 mV/°C
- MEA832 per l'umidità con caratteristica seguente: (1)
- NORP12 per la luminosità con caratteristica seguente: (2)



caratt. MEA832 (1)



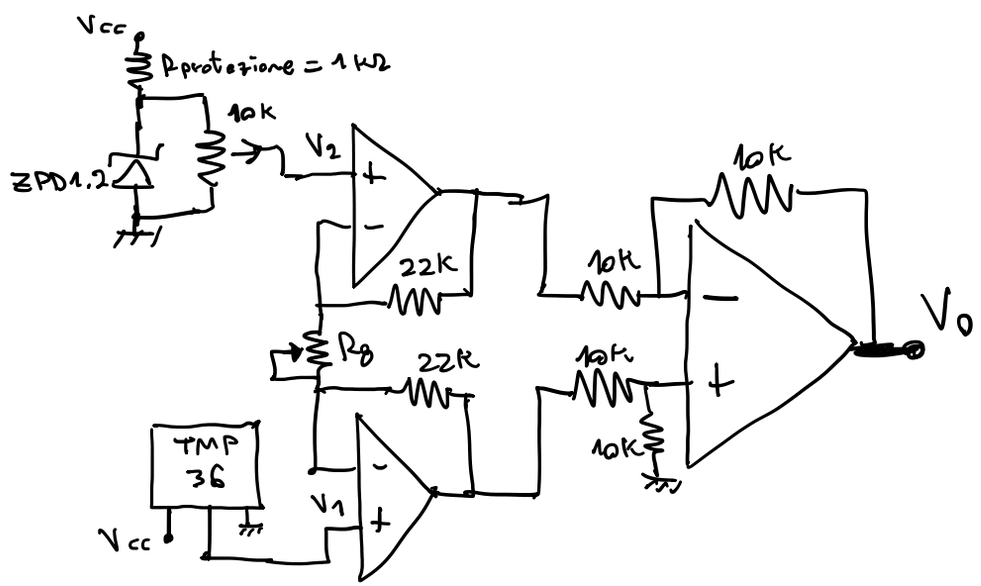
caratt. NORP12 (2)

(3)

la risoluzione dell'ADC di arduino è di 4,9 mV, dato che il trasduttore TMP36 varia la sua tensione di uscita di 10 mV/°C la risoluzione finale è di 0,5 °C, potenzialmente sufficente ma comunque bassa. Al fine di ottenere una risoluzione ottimale si può utilizzare un circuito di condizionamento come l'amplificatore differenziale per strumentazione. I vantaggi di questo tipo di configurazione rispetto a un differenziale standard è l'elevata precisione garantita dalla realizzazione chimica delle resistenze integrate che per questo motivo hanno valore identico. NB: realizzato a componenti discreti per Tinkercad (non c'è l'INA111)

Scelgo un range di temperatura dato da due estremi, il primo minore della temperatura minima invernale e il secondo maggiore della temperatura massima estiva → [0°C ; 25°C]
Applicando una tensione continua su V_2 questa viene sottratta dal valore di V_1. Scegliendo V_2 = 500 mV, ovvero alla V_{out} a 0°C del TMP36, a 0°C ottengo 0V in uscita dal circuito di condizionamento. Per ottenere 5V in uscita alla temperatura massima (25°C) al fine di massimizzare l'amplificazione, conoscendo la V_{out} del TMP36 a 25°C (250 mV con offset V_2 già applicato) è sufficente applicare un amplificazione data da

$$\frac{V_{max\,ADC}}{V_1 - V_2}, \text{ ovvero } \frac{5}{250 \cdot 10^{-3}} = 20$$



- Per realizzare l'offset (V_2) costante da 500 mV è sufficiente realizzare un partitore di tensione con potenziometro su un diodo zener con V_{ZK} = 1,2V. La scelta di utilizzare un potenziometro come partitore è data dalla possibilità di calibrare precisamente la tensione in uscita dal contatto mobile.

- La resistenza di protezione del diodo zener serve a limitare la corrente che può scorrere in quest'ultimo. 1kΩ permette il passaggio di 10,8 mA (con V_{cc} = 12V), quindi > 5 mA, ovvero la corrente minima per permettere al diodo zener di funzionare da stabilizzatore, e < 50 mA, ovvero la corrente massima supportata.
- Successivamente, è necessario calcolare il valore di R_g, dato da:

$$V_o = (V_1 - V_2)\left(\frac{2R}{R_g} + 1\right)$$

$$5 = 250 \cdot 10^{-3}\left(\frac{44 \cdot 10^3}{R_g} + 1\right)$$

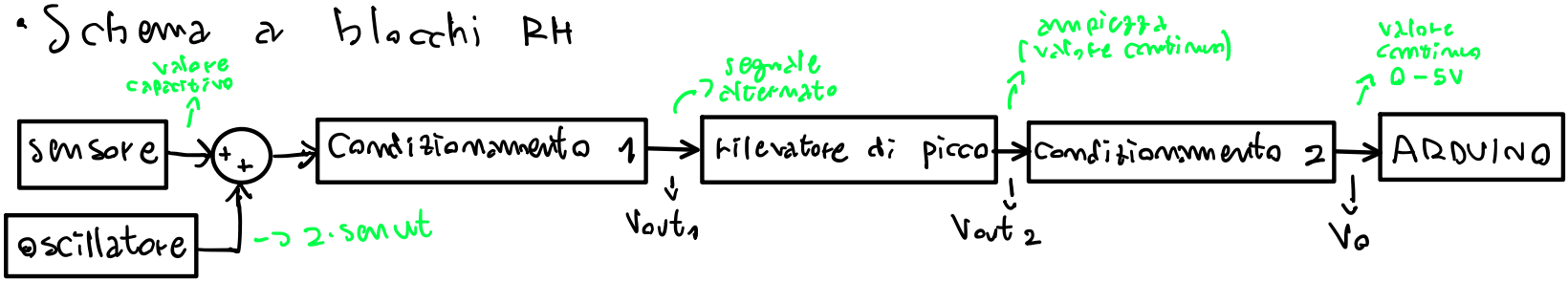
$$\frac{5}{250 \cdot 10^{-3}} = \frac{44 \cdot 10^3}{R_g} + 1$$

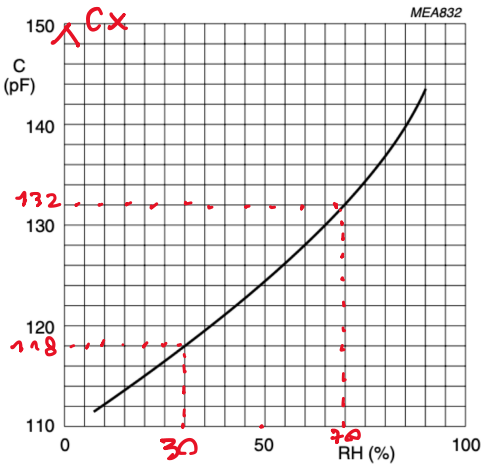
$$R_g = \frac{44 \cdot 10^3 \cdot 250 \cdot 10^{-3} + 250 \cdot 10^{-3}}{5} = 2{,}2\,k\Omega$$

Anche in questo caso R_g è scelto come potenziometro (3,3kΩ) in modo da poter essere regolato precisamente a 2,2kΩ

Per quanto riguarda il sensore di umidità relativa MEA832 non è necessario sfruttare l'intera caratteristica di trasferimento, dato che RH deve essere compresa tra il 40% e il 50%. Per questo motivo scelgo un range dal 30% al 70%, in cui la caratteristica è pressoché lineare.

- Schema a blocchi RH





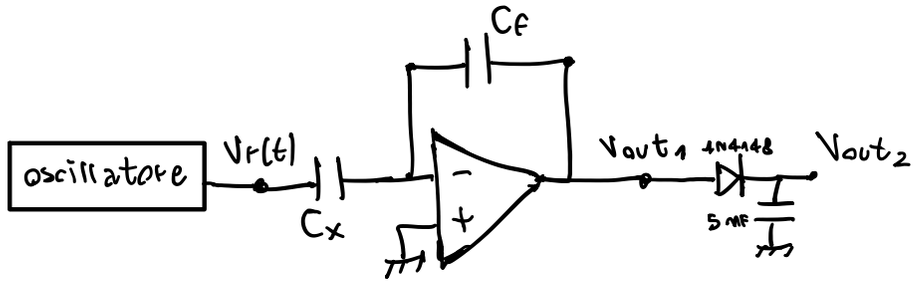
Al fine di convertire il valore capacitivo del sensore in un segnale in tensione alternato con ampiezza pari a V_{out1} utilizzo il seguente circuito. Dato che ultimamente sarà necessario ottenere un valore continuo sarà posizionato in cascata al seguente circuito un raddrizzatore a singola semionda (o rilevatore di valore di picco).

Scelgo CF come un valore confrontabile con Cx (100pF)

Scelgo l'ampiezza del segnale $V_r(t)$ come 3V (circa metà del valore di tensione massimo supportato dall'ADC di ARDUINO UNO.

Scelgo la frequenza del segnale oscillatorio in modo che l'impenza Z del condensatore di retroazione (CF) sia di qualche kΩ

$$Z = \frac{1}{2\pi f C} \rightarrow f = \frac{1}{2\pi \cdot Z \cdot C}, \quad \text{con } Z = 10\,k\Omega \rightarrow f \simeq 150\,kHz$$

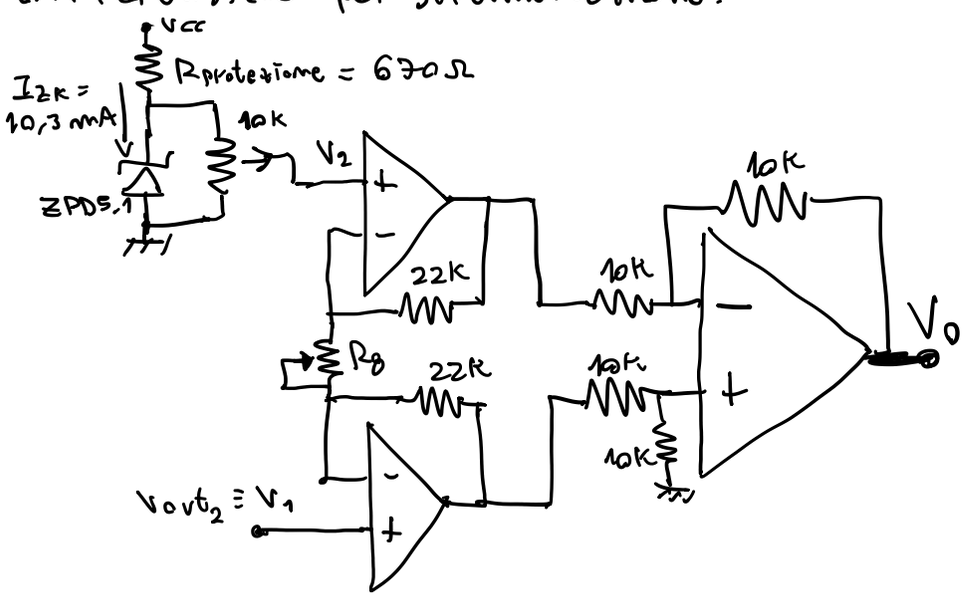


$$\begin{cases} RH = 30\% & V_{out1\,max} = \frac{C_x}{C_f} \cdot V_{r\,max} = \frac{118}{100} \cdot 3 = 3{,}54\ V \\ RH = 70\% & V_{out1\,max} = \frac{C_x}{C_f} \cdot V_{r\,max} = \frac{132}{100} \cdot 3 = 3{,}96\ V \end{cases}$$

I valori di V_{out2} saranno corrispondenti a $V_{out1} - 0{,}6V$ dati dalla tensione di polarizzazione del diodo, ovvero:

$$\begin{cases} RH = 30\% & V_{out2} = V_{out1\,max} - 0{,}6 = 2{,}94\ V \\ RH = 70\% & V_{out2} = V_{out1\,max} - 0{,}6 = 3{,}36\ V \end{cases}$$

Al fine di adattare questi valori all'ADC di arduino, quindi ottenere $V_o\,[0;5V]$ per $RH\,[30\%;\ 70\%]$ è possibile utilizzare nuovamente l'amplificatore differenziale per strumentazione.



- L'offset (V_2) in questo caso è 2,94 V, realizzato come nel caso del TMP36 però partendo da una tensione di tettura $V_{zk} = 5{,}1V$

- Calcolo R_g

$$V_o = (V_1 - V_2)\left(\frac{2R}{R_g} + 1\right)$$

$$5 = 420 \cdot 10^{-3}\left(\frac{44 \cdot 10^3}{R_g} + 1\right)$$

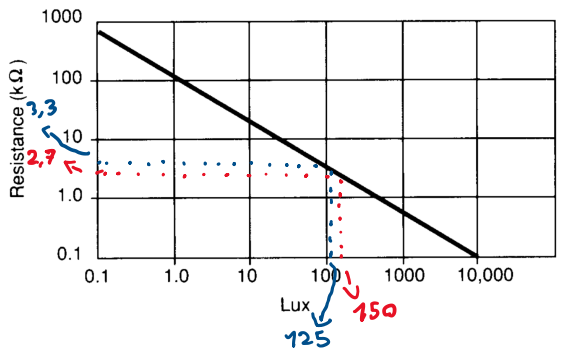
$$\frac{5}{420 \cdot 10^{-3}} = \frac{44 \cdot 10^3}{R_g} + 1$$

$$R_g = \frac{(44 \cdot 10^3 \cdot 420 \cdot 10^{-3}) + 420 \cdot 10^{-3}}{5} = 3{,}7\,k\Omega$$

scelgo potenziometro da 4,7 kΩ

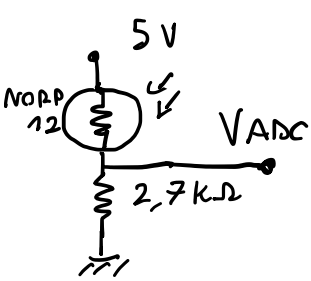
Per quanto riguarda il rilevamento dell'intensità luminosa interna, osservando la caratteristica di trasferimento del fotoresistore NORP12 è possibile trovare il suo valore di resistenza in presenza di 150 lux, ovvero circa 2,7 kΩ.

Figure 4 Resistance as a function illumination



→ NB: Il diagramma è in scala logaritmica su entrambi gli assi.

Si presuppone che il sito archeologico sia sotterraneo perciò al chiuso. Per questo motivo la luminosità dell'ambiente è data interamente dalle lampade. Per ogni stanza ci sono 6 lampade e un ARDUINO posto al centro. Grazie a un semplice circuito pilotato da ARDUINO UNO è possibile rilevare e correggere il valore di luminosità interna.



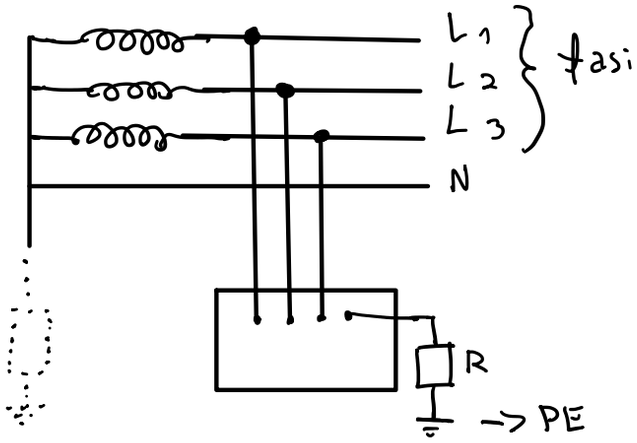
- L'illuminazione è composta da 6 lampade da 25 lux per un totale di 150 lux. In questa condizione ARDUINO legge un valore in ingresso sull'ADC pari a 2,5V. Nel caso in cui una lampada smetta di funzionare ARDUINO accende una settima lampada "di emergenza" e accende un LED rosso per segnalare il problema. Nel caso pratico in cui si rompe una lampada, e dunque la luminosità scende a 125 lux, V_{ADC} non sarà più 2,5V ma sarà:

$$\frac{2,7\cdot 10^3}{2,7\cdot 10^3 + R_{NORP}}\cdot 5 = \frac{2,7\cdot 10^3}{2,7\cdot 10^3 + 3,3\cdot 10^3}\cdot 5 = 2,25V$$

Dunque, è sufficiente programmare ARDUINO affinché, quando la tensione in ingresso scenda sotto 2,3V proceda con la procedura "di emergenza" precedentemente descritta.

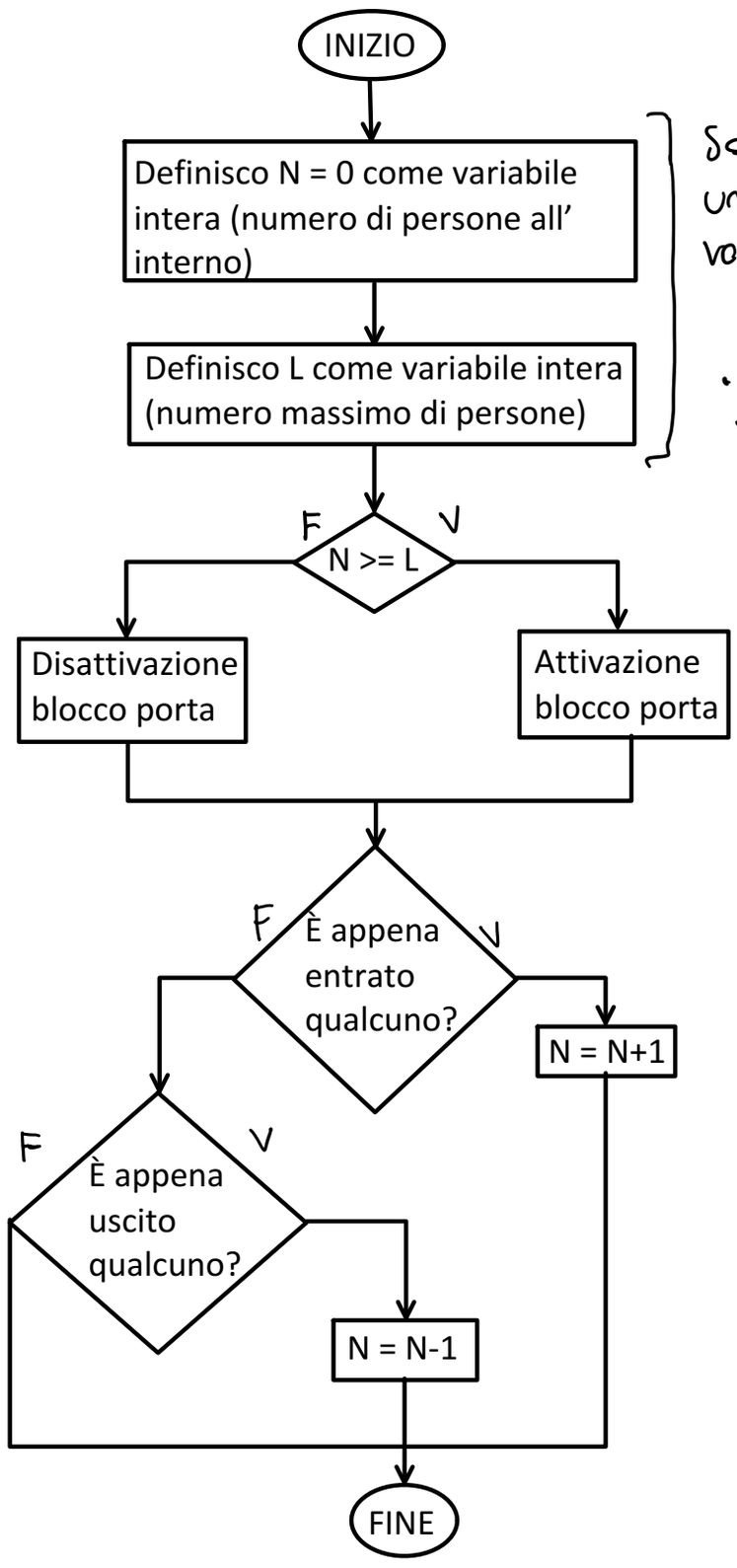
Per quanto riguarda l'impianto elettrico, data l'importanza della continuità del servizio elettrico, il sistema IT (Isolato da Terra) rappresenta la soluzione ottimale, in conformità con la normativa CEI 64-15 specifica per gli edifici pregevoli per arte e storia. Questo sistema garantisce infatti il funzionamento in caso di primo guasto a terra, permettendo ai sistemi di controllo di continuare a funzionare. È possibile introdurre anche un sistema di accumulo al fine di garantire continuità anche in caso di disservizio della rete elettrica. È necessario implementare anche l'interruttore magnetotermico e differenziale (in questo caso utile soltanto in caso di doppio guasto a terra, data la scelta del sistema IT). Il magnetotermico proteggerà da eventuali sovraccarichi e cortocircuiti

- Schema sistema IT



4

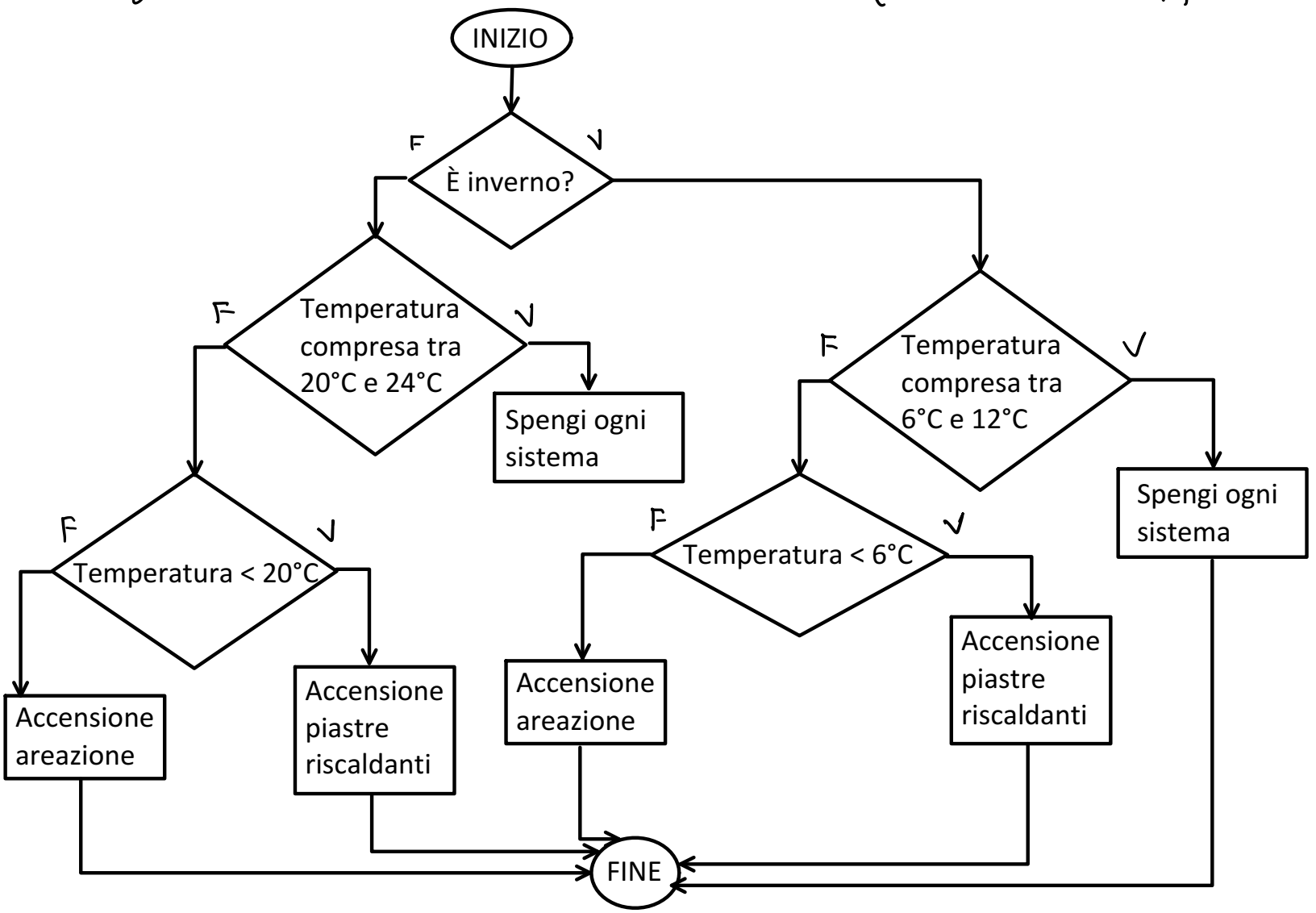
• Diagramma di flusso controllo tornelli (ripetuto in loop)



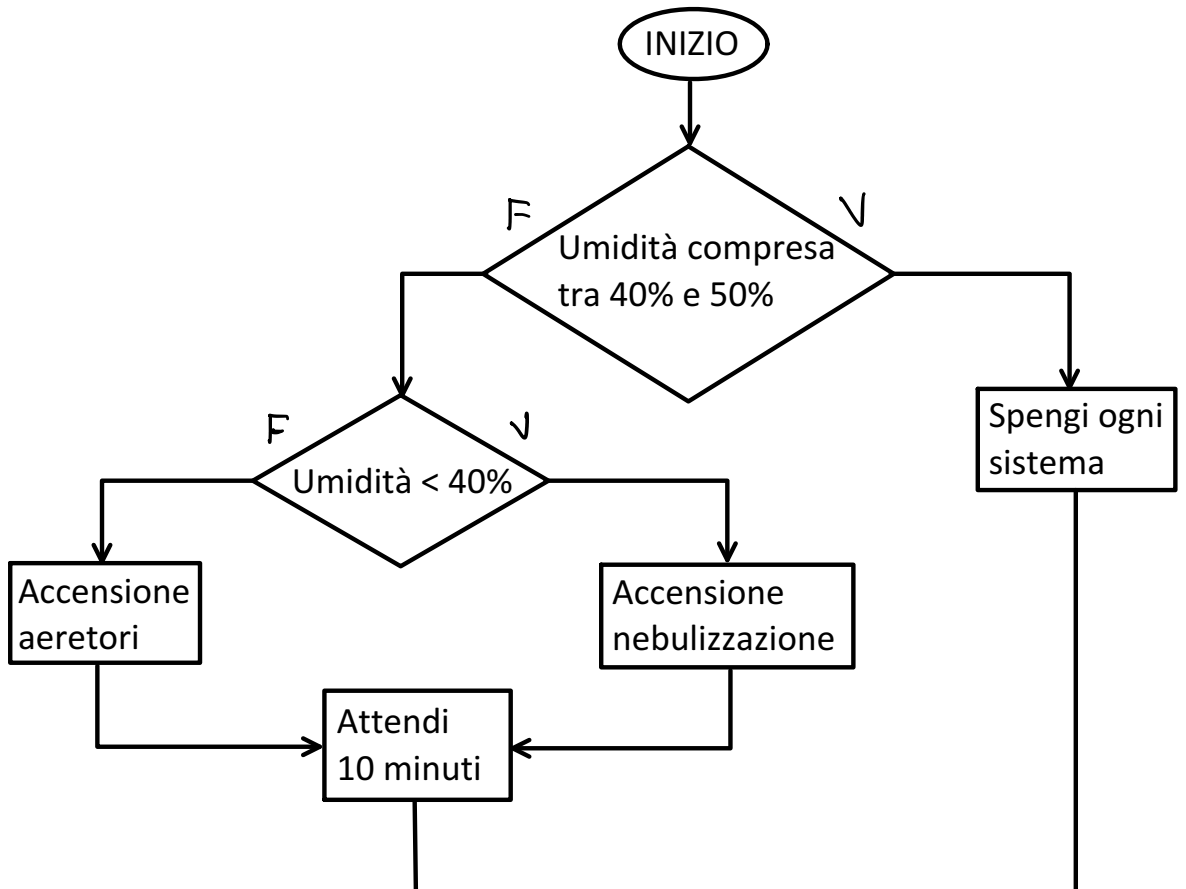
Solo una volta

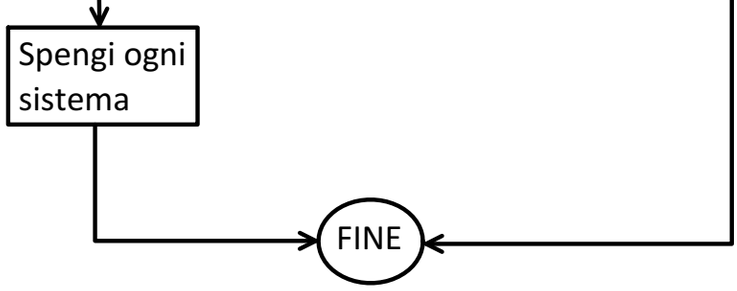
• Il valore L (limite) varia in base alla stanza in cui è posizionato; per i locali affrescati corrisponde a 5, mentre per i tornelli di ingresso principali corrisponde a 30.

• Diagramma di flusso controllo temperatura (ripetuto in loop)



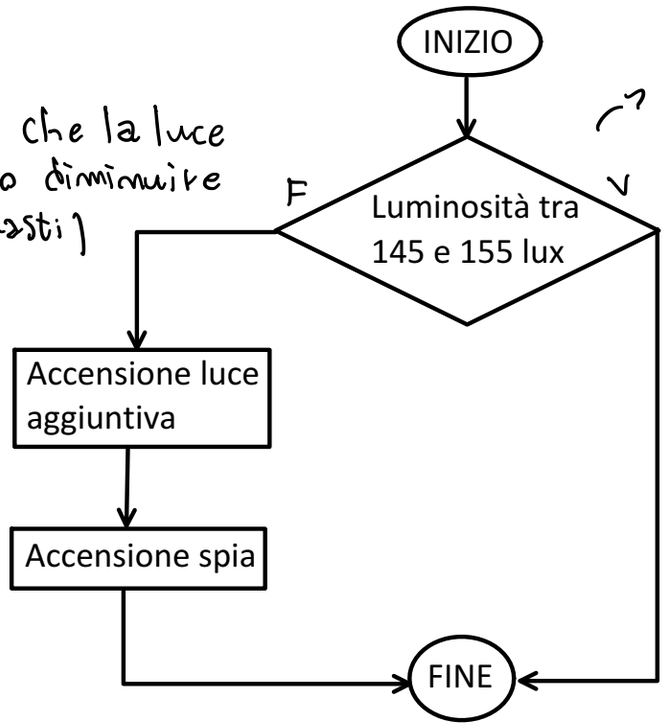
• Diagramma di flusso controllo umidità (ripetuto in loop)





• Diagramma di flusso controllo luminosità (ripetuto in loop)

Si presuppone che la luce possa soltanto diminuire (a causa di guasti)



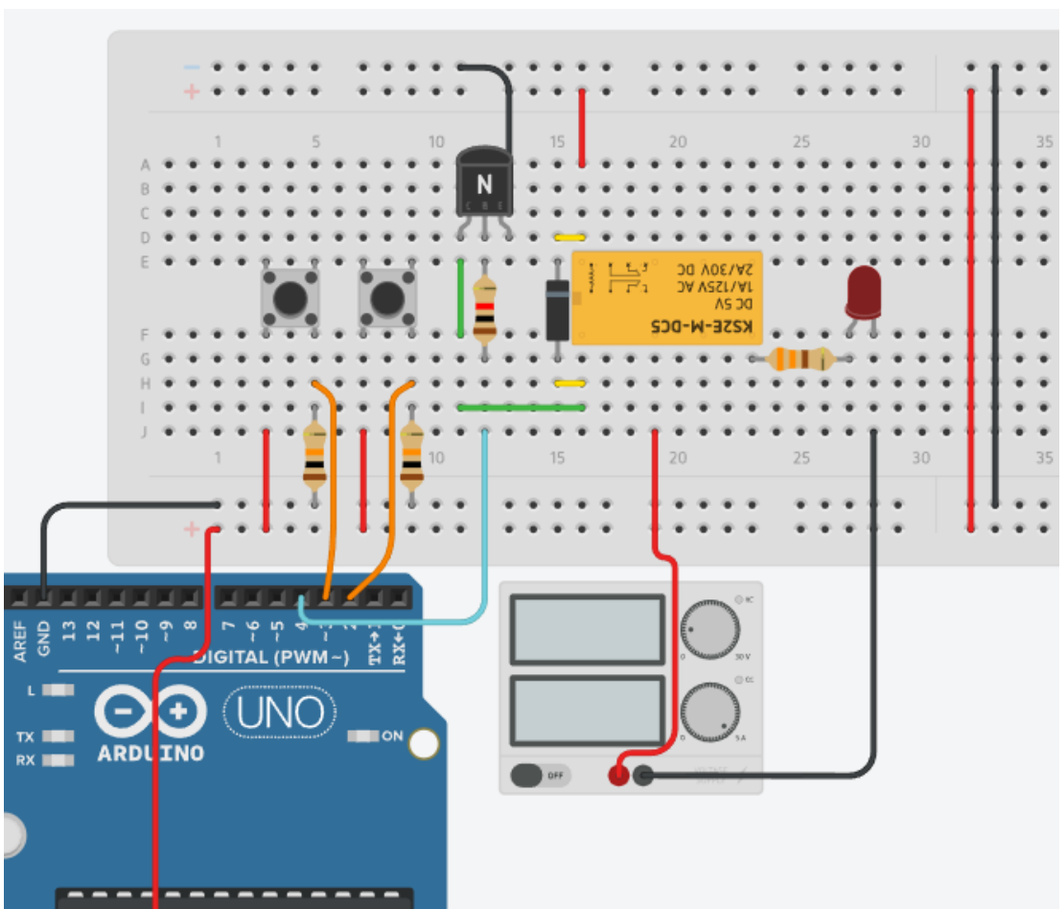
Nel testo non è specificato un range specifico, scelgo ±5 lux dai 150 indicati

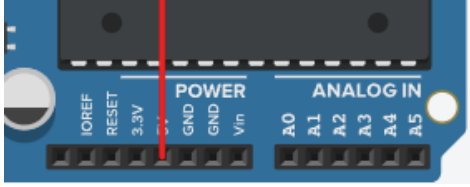
Per disattivare la spia, una volta sostituita la lampada guasta, è sufficiente riavviare il sistema di controllo

5

• Tinkercad e codici ARDUINO

- Controllo tornelli

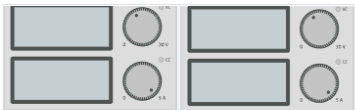


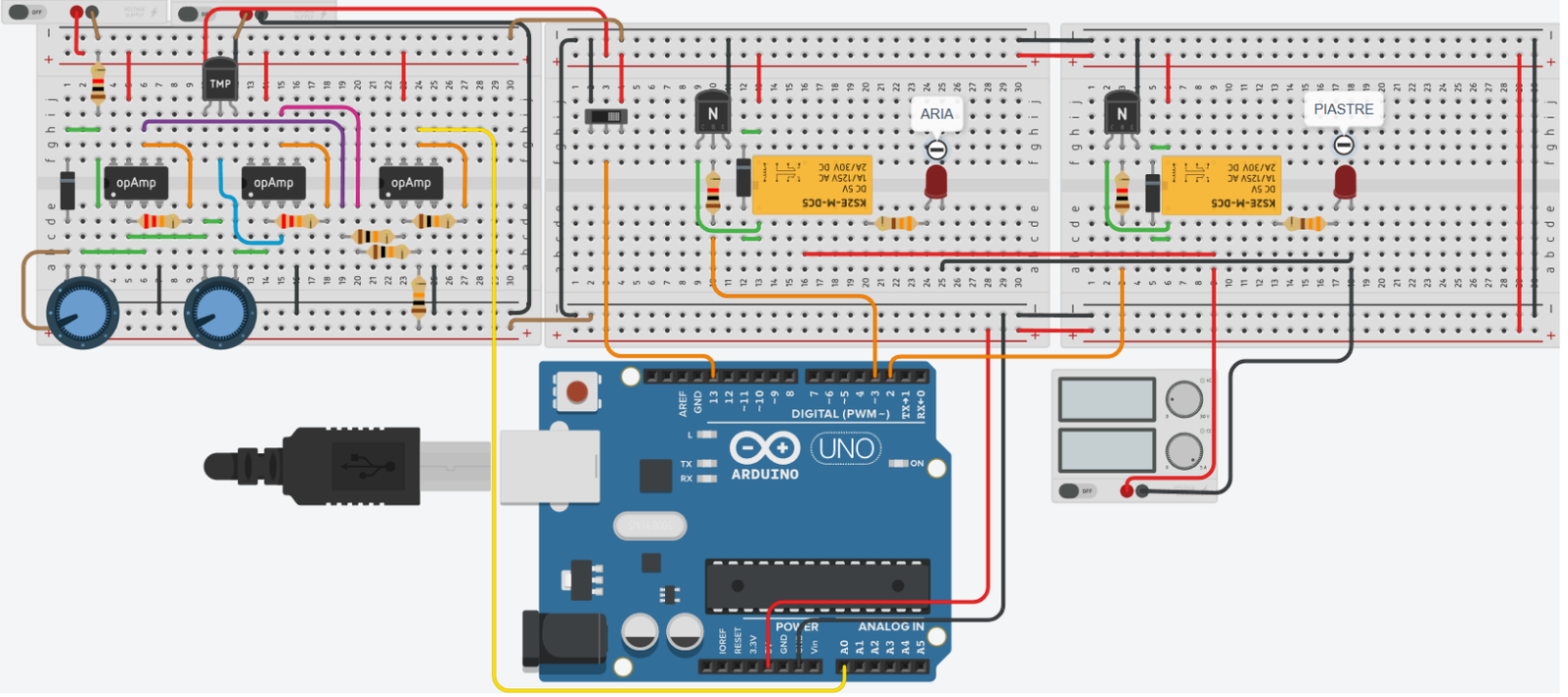



```
int entra = 3;
int esci = 2;
int led = 4;
int persone = 0;
int limite = 5;
bool stopconto = false;
bool stopconto2 = false;
void setup()
{
  pinMode(entra, INPUT);
  pinMode(esci, INPUT);
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
 // return;
  bool statoentra = digitalRead(entra);
  bool statoesci = digitalRead(esci);
  if ((statoentra == true) && (stopconto == false))
  {
    persone += 1;
    stopconto = true;
  }
  if (statoentra == false)
  {
    stopconto = false;
  }
  if ((statoesci == true) && (stopconto2 == false))
  {
    persone -= 1;
    stopconto2 = true;
  }
  if (statoesci == false)
  {
    stopconto2 = false;
  }
  if (persone >= limite)
  {
    digitalWrite(led, true);
  }
  else
  {
    digitalWrite(led, false);
  }
  Serial.print("persone: ");
  Serial.println(persone);
  delay(100);
}
```

- Controllo temperatura



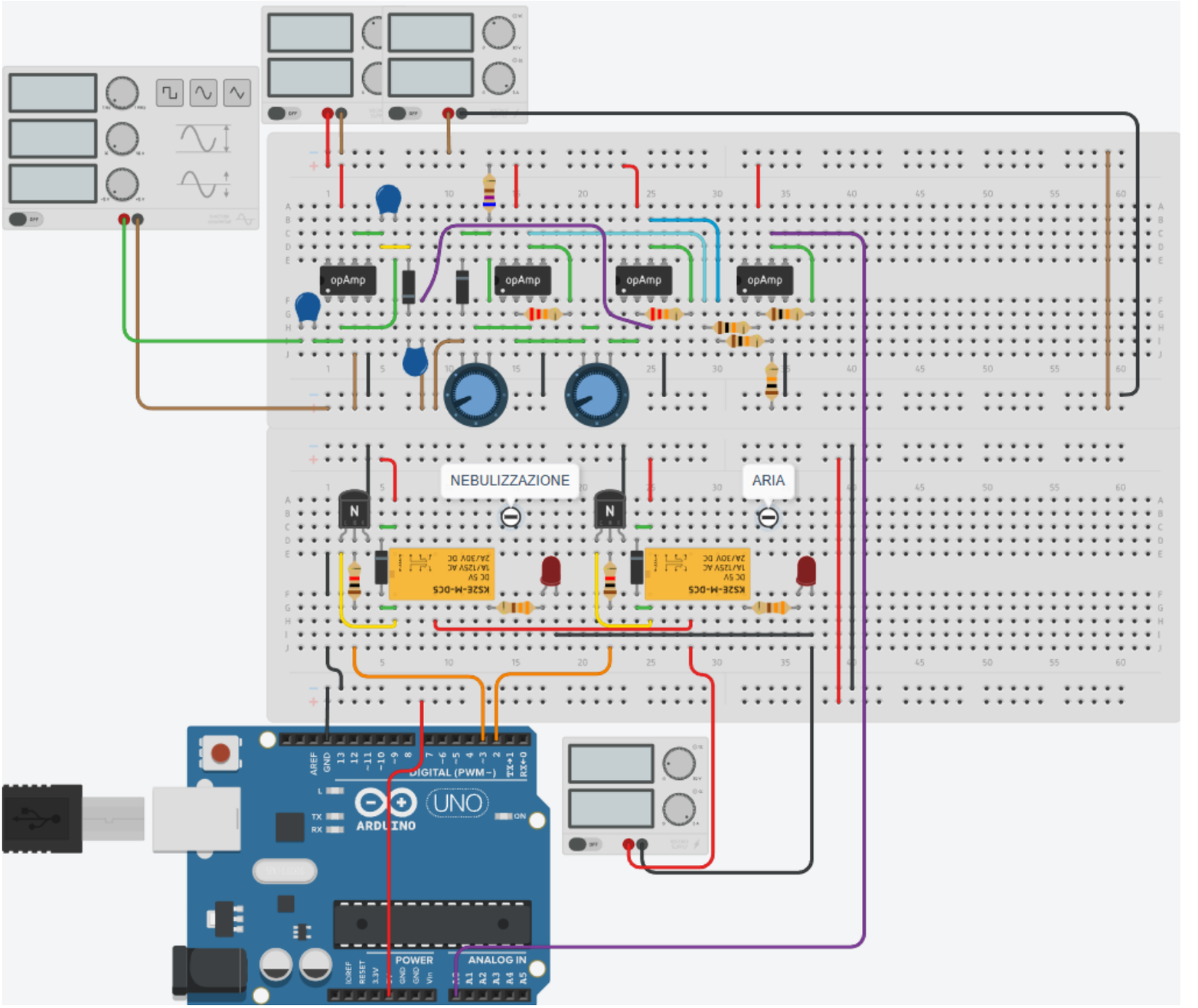



```
int interruttore = 13;
int tmp = A0;
int piastre = 2;
int aria = 3;
void setup()
{
  pinMode(interruttore, INPUT);
  pinMode(piastre, OUTPUT);
  pinMode(aria, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  bool statoint = digitalRead(interruttore);
  int adc = analogRead(tmp);
  float tensionemv = adc/0.2048;
  float temperatura = tensionemv/200;
  Serial.println(tensionemv);
  Serial.println(temperatura);
  if (statoint == false)
  {
    Serial.println("Mod. Invernale");
    if (temperatura <= 6.0)
    {
      digitalWrite(piastre, true);
      digitalWrite(aria, false);
    }
    if ((temperatura >= 8.5) && (temperatura <= 9.5))
    {
      digitalWrite(piastre, false);
      digitalWrite(aria, false);
    }
    if (temperatura >= 12.0)
    {
      digitalWrite(aria, true);
      digitalWrite(piastre, false);
    }
  }
  else
  {
    Serial.println("Mod. Estiva");
    if (temperatura <= 20.0)
    {
      digitalWrite(piastre, true);
      digitalWrite(aria, false);
    }
    if ((temperatura >= 21.5) && (temperatura <= 22.5))
    {
      digitalWrite(piastre, false);
      digitalWrite(aria, false);
    }
    if (temperatura >= 24.0)
    {
```

```
      digitalWrite(aria, true);
      digitalWrite(piastre, false);
    }
  }
  delay(100);
}
```

• Controllo umidità

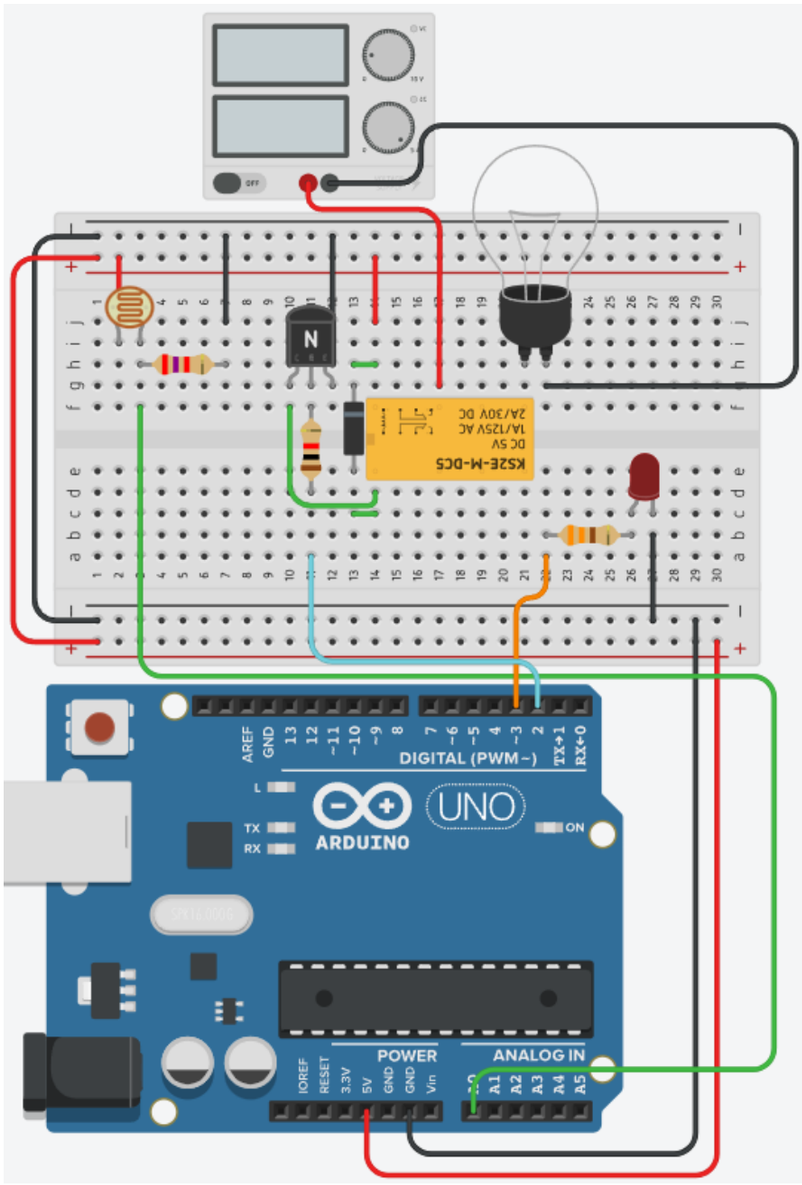



```
int sens = A0;
int nebulizzazione = 3;
int aria = 2;
int tempoms = 600000;
void setup()
{
  Serial.begin(9600);
  pinMode(sens, INPUT);
  pinMode(aria, OUTPUT);
  pinMode(nebulizzazione, OUTPUT);
}

void loop()
{
  int adc = analogRead(sens);
  float tensionemv = adc/0.2048;
  float umidita = (tensionemv/125) + 30;
  Serial.println(umidita);
  if (umidita < 40)
  {
```

```
    Serial.println("nebulizzazione abilitata");
    digitalWrite(nebulizzazione, HIGH);
    delay(tempoms);
    digitalWrite(nebulizzazione, LOW);
  }
  if (umidita > 50)
  {
    Serial.println("aria abilitata");
    digitalWrite(aria, HIGH);
    delay(tempoms);
    digitalWrite(aria, LOW);
  }
  delay(100);
}
```

- Controllo luminosità




```
int sens = A0;
int lampada = 2;
int spia = 3;
void setup()
{
  pinMode(sens, INPUT);
  pinMode(lampada, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int adc = analogRead(sens);
  float tensionemv = adc/0.2048;
  if (tensionemv < 2300)
  {
    digitalWrite(lampada, HIGH);
    digitalWrite(spia, HIGH);
  } else {
    digitalWrite(lampada, LOW);
  }
  Serial.println(tensionemv);
  delay(100);
}
```