MONITORAGGIO LIEVITAZIONE DELL'IMPASTO PER PIZZA

L'OBBIETTIVO

Mantenere stabile la temperatura ambiente per favorire la lievitazione dell'impasto per pizza

PREMESSE GENERALI

Per una corretta lievitazione l'impasto deve riposare per un lungo periodo di tempo (anche 24h) ad una temperatura il più possibile costante che varia dai 24°Celsius ai 32°Celsius, con un'umidità relativa tra il 70% ed 80%.

La temperatura verrà garantita sfruttando i principii dell'effetto Joule per cui ogni metallo attraversato da una corrente elettrica cede energia sottoforma di calore.

REQUISITI DEL PROGETTO

Basso consumo

Considerate le tempistiche di una lievitazione, si devono garantire bassi consumi, calibrando gli attuatori e garantendo una corretta coibentazione.

Taratura sensori

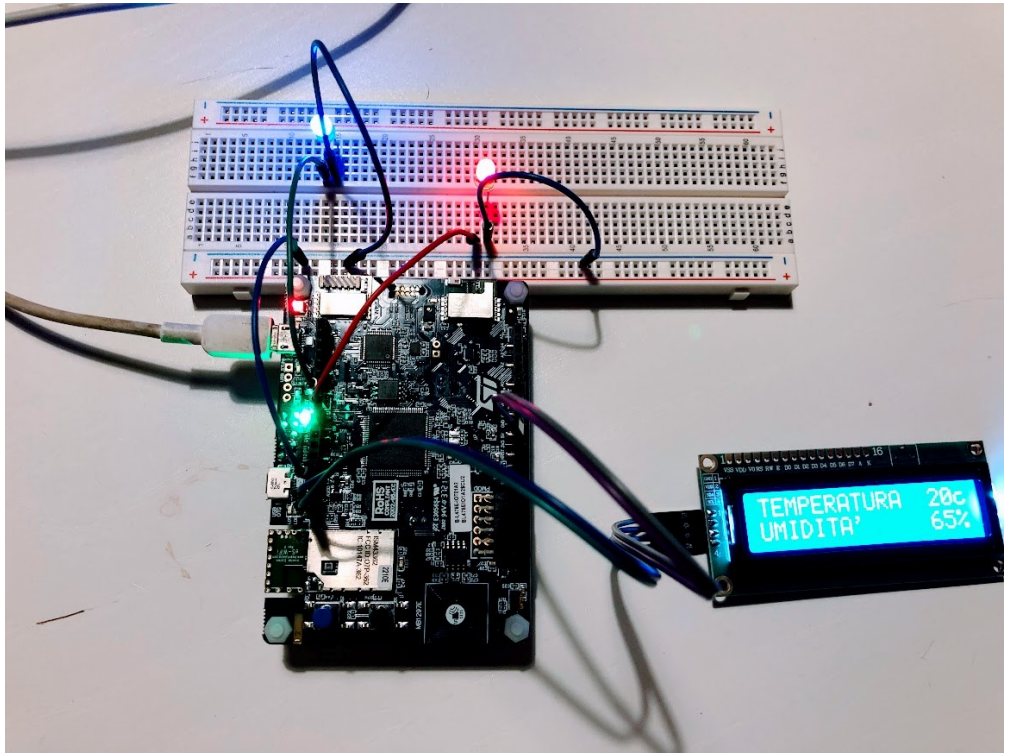
Visto che lettura del dato potrebbe essere influenzata dalla scheda stessa, diventa indispensabile tarare correttamente il sensore di temperatura.

User friendly

Lo strumento deve lavorare in autonomia fornendo i dati in modo semplice, veloce e chiaro.

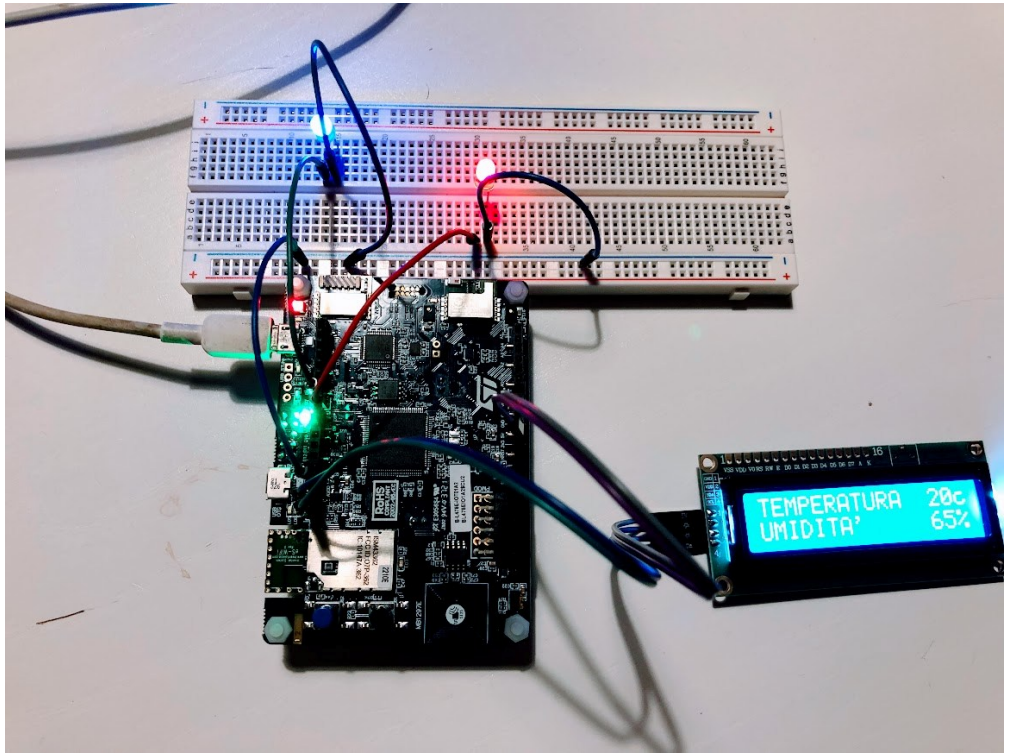
IL PROTOTIPO

Per il progetto è stata utilizzata la scheda di sviluppo Stm32 b-l475e-iot01a2: garantisce bassi consumi, ha un sensore di temperatura e umidità integrato, uscite digitali per comandare gli attuatori e interfacce I2C per il collegamento del display. Inoltre è fornita di scheda wifi che permette un'espansione del progetto a livello di domotica.



IL PROTOTIPO

Il progetto prevede che il sensore di temperatura e di umidità a bordo della scheda raccolga i dati dell'ambiente. I dati vengono mandati in output sul display digitale. Il microcontrollore si occuperà dell'elaborazione e, se il valore di temperatura di umidità scende sotto una data soglia, porta a livello logico 1 l'uscita digitale d7 collegata al led blu (attuatore termico) e in toggle l'uscita digitale d8 (allarme).



COMPONENTI FONDAMENTALI

- Sensore di umidità/temperatura HTS221: ideale perché ha un range operativo tra i -40 e 120°Celsius con una buona linearità e precisione tra i 15 e 40°Celsius
- Display Freenova lcd1602 16x2 con modulo I2c integrato. Operativo fino a 55°Celsius

HLGA-6L
(2 x 2 x 0.9 mm)

SVILUPPO COMMERCIALE

Per lo sviluppo commerciale del progetto occorre:

- **Ambiente coimbentato** di misura 0.50 x 0.38 x 0.40 mt con un volume di 0,076 m^3
- **Attuatore termico** con una potenza minima in watt così calcolata

  V*dT*K=Potenza termica espressa in cal/h

  Temperatura media casalinga: 20°C

  Temperatura da raggiungere: 30°C

  Coefficiente termico K polistirolo: 1 (eccesso)

  0,076*10*1=760 cal/h→0,88 Watt di potenza necessarie.

SVILUPPO COMMERCIALE

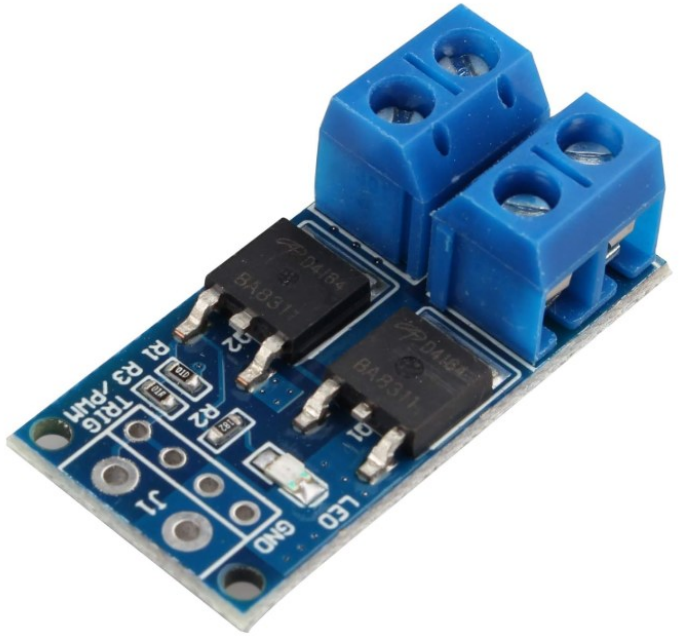
Attuatore termico

Riscaldatore a semiconduttore RC 016 con potenza massima di 13watt, range di tensione in ingresso che va dai 12 ai 30 Volt, capace di sopportare temperature fino ai 170 gradi celsius e ip32.

SVILUPPO COMMERCIALE

Per pilotare l'accensione dell'attuatore termico sarà necessario un Trigger Switch Drive Module (modello ICQUANZX DC 5V-36V).

L'uscita D7 della scheda (attualmente collegata al led blu) piloterà la tensione ai gate per permettere ad una DC compresa tra i 5 e 36v di passare ad alimentare l'attuatore.



SVILUPPO COMMERCIALE

Trasformatore DC

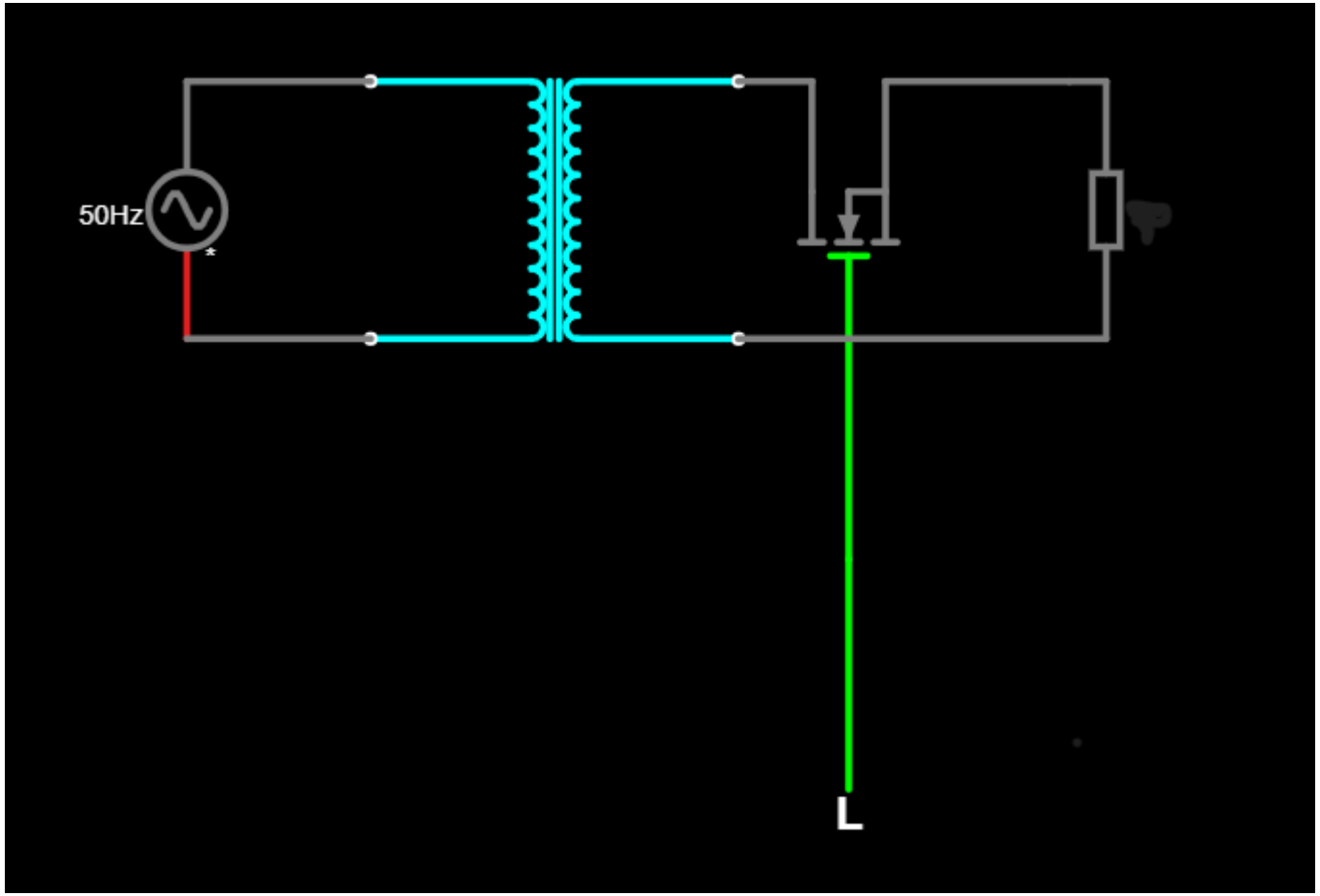
L'attuatore termico scelto ha una potenza massima di 13 Watt con un tensione in input che può variare da 12 a 30 volt.

Quindi dato che $P = \frac{V^2}{R}$ e $R = \frac{V^2}{P}$

ricaviamo che idealmente il riscaldatore ha un resistenza di 69Ω.

Quindi si ricava che con un alimentatore da 12volt si raggiungono i 2watt di potenza.

SVILUPPO COMMERCIALE



STRUTTURA DEL CODICE

Inizializzazione

Dopo aver inizializzato i vari pin, ho riempito tutti i 32 spazi del display e acceso tutti i led per testarne il corretto funzionamento.

```
MX_GPIO_Init();
MX_USART1_UART_Init();
MX_I2C1_Init();
HAL_UART_Init(&huart1);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);//test led
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 1);// test led
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);//pin attuatore termico
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);// pin allarme

HD44780_Init(2);
HD44780_Clear();
HD44780_SetCursor(0,0);
HD44780_PrintStr("Lievito********");//controllo schermo riga 1
HD44780_SetCursor(0,1);
HD44780_PrintStr("*************2.0");//controllo schermo riga 2
HAL_Delay(3000);
HD44780_Clear();
HD44780_SetCursor(0,0);
HD44780_PrintStr("TEMPERATURA");
HD44780_SetCursor(0,1);
HD44780_PrintStr("UMIDITA'");
HD44780_SetCursor(15,0);
HD44780_PrintStr("c");
HD44780_SetCursor(15,1);
HD44780_PrintStr("%'");
BSP_HSENSOR_Init();
BSP_TSENSOR_Init();
```

STRUTTURA DEL CODICE

Loop

- Vengono assegnate i valori rilevati dai sensori alle variabili h e t, calibrando il dato di quest'ultima e stampando in output sul display
- Viene impostato il valore di controllo alla variabile c ed effettuato il controllo: se t è inferiore a c d7 accenderà il led attuatore e d8 eseguirà il toogle sul led allarme

```
while (1)
      {

      c=30;//temperatura di controllo
      h = BSP_HSENSOR_ReadHumidity();//leggo umidità
      t = BSP_TSENSOR_ReadTemp();//leggo temperatura
      t = t-1; //correzione temperatura percepita con ambientale

      //mando in output i valori rilevati
      HAL_Delay (500);
      //trasmissione su display
      itoa(t, s, 10); //converti in stringa intero temperatura
      HD44780_SetCursor(13,0);
      HD44780_PrintStr(s); //stampa temperatura su display
      itoa(h, s, 10); //converti in strg int umidità
      HD44780_SetCursor(13,1);
      HD44780_PrintStr(s);//stamapa umidità display
      //trasmissione via seriale



      if (t<c)//se temp è minore di controllo
        {
         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);//tieni resistore e allarme a 1
         //HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 1);
         HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_2);
         HAL_Delay(2000);
         }
          else //altrimenti mantieni lo stato a 0, se non lo metto potrebbe sfalsare valori
          {
          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);//tieni resistore e allarme a 0
          HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);
          HAL_Delay(2000);
          }

          }
```