

Aggiornamento

Applicazioni dei sensori

Impiego del sensore DHT11 per misure di temperatura e umidità

Il sensore DHT11 viene utilizzato per effettuare misure di temperatura e umidità ambientale fornendo un segnale digitale con valore che dipende da queste due grandezze fisiche.

Per misurare l'umidità viene utilizzato un metodo resistivo, per la temperatura un NTC.

La tensione generata dai due sensori viene convertita in formato digitale da un convertitore A/D a 8 bit.

All'interno di questo sensore, oltre al convertitore, è presente anche un microcontrollore che contiene i coefficienti di calibrazione per la misura della temperatura e dell'umidità.

Il sensore (figura 1) viene montato su una scheda appositamente predisposta e facilmente reperibile in commercio.

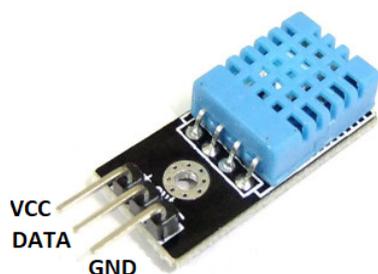


Figura 1

Il sensore presenta 3 pin:

- VCC (per la tensione di alimentazione positiva);
- GND (per la massa);
- DATA (per la trasmissione del segnale).

Il pinout viene di solito stampato sulla scheda.

Il sensore può essere alimentato con tensioni comprese fra 3 V e 5,5 V.

La corrente assorbita dal sensore ale:

- circa 1,5 mA se sta misurando il dato di temperatura o umidità;
- circa 120 uA se è in standby ovvero quando aspetta il segnale di start dal microcontrollore.

Le caratteristiche di misura dei due sensori vengono riportate nella tabella 1.

	UMIDITA'	TEMPERATURA
Intervallo di misura	20-90 %	0-50 °C
Risoluzione	1 %	1 °C
Tempo di risposta	10 s	13 s

Tabella 1

Nello schema per la connessione tra una scheda con microprocessore e il sensore riportato in figura 2 si evidenzia la presenza di un resistore di pull-up tra linea dati e alimentazione.

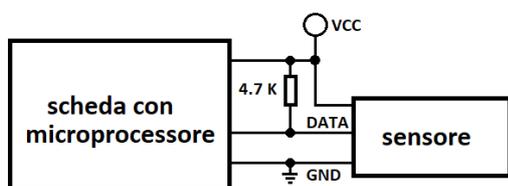


Figura 2

Per la comunicazione viene usato un formato a singolo bus. L'intervallo di tempo necessario per effettuare una comunicazione è di circa 4 ms. Ciascun dato è formato da 40 bit ovvero:

- 8 bit per il dato di umidità integrale;
- 8 bit per la parte decimale del dato di umidità
- 8 bit per il dato di temperatura integrale;
- 8 bit per la parte decimale del dato di temperatura;
- 8 bit di controllo.

Per avviare la comunicazione il microprocessore deve inviare il segnale di start; il sensore esce dalla modalità standby e inizia a comunicare con il microprocessore; al termine della comunicazione ritorna in standby.

La temporizzazione dei segnali della linea DATA che consente l'avvio della comunicazione viene mostrata in figura 3.

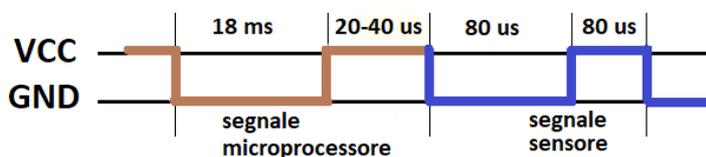


Figura 3

Se non si ha comunicazione il bus rimane a livello logico alto.

Quando il microprocessore vuole avviare una comunicazione opera nel modo seguente:

- esegue una transizione da livello alto a livello basso;
- mantiene il livello logico basso per 18 ms;
- si porta a livello logico alto e attende per 20-40 us la risposta del sensore.

Il sensore risponde nel modo seguente:

- porta il bus a livello logico basso per 80;
- porta il bus a livello logico alto per ulteriori 80 us.

Di seguito parte la comunicazione dei dati ovvero la trasmissione dei 40 bit.

I bit 0 e 1 vengono identificati con un codice temporale nel modo seguente:

- lo 0 logico con uno stato alto del bus della durata di 26-28 us;
- l'1 logico con uno stato alto del bus di durata 70 us.
- tra un bit e l'altro viene interposto un livello logico basso di 50 us.

Anche l'ultimo bit del dato è seguito da uno stato basso del bus di durata 50 us; successivamente il bus viene mandato a livello logico alto per la presenza del resistore di pull-up.
 Un esempio viene riportato in figura 4.

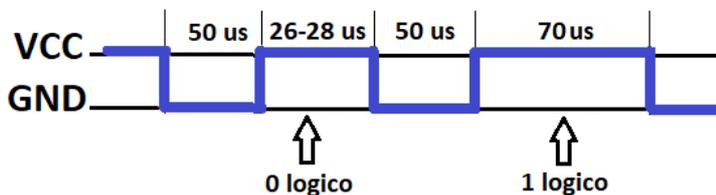


Figura 4

Esempio di impiego con la scheda Arduino

Viene di seguito fatto riferimento al sensore DHT11 montato su una scheda commercializzata da dfrobot; chi non ha a disposizione la scheda può effettuare i collegamenti in modo analogo.

Per il collegamento tra la scheda del sensore e la scheda Arduino (fig.) vengono utilizzati tre cavetti; due devono essere collegati rispettivamente all'alimentazione VCC (+5 V) e al GND; il terzo terminale (quello di segnale per il trasferimento dei dati) deve essere collegato ad uno dei pin digitali della scheda (in questo caso al pin 2).

Si osservi per i collegamenti la figura 5.

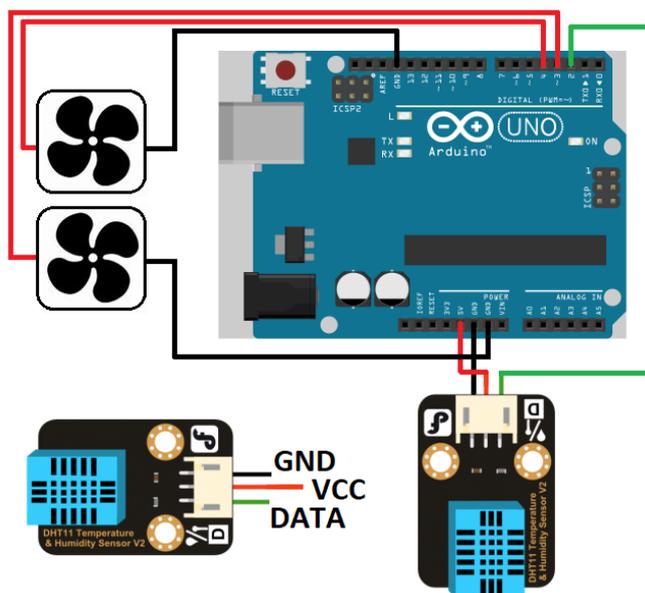


Figura 5

La programmazione risulta facilitata per la presenza di una libreria apposita facilmente reperibile sul web (la dht11.h).

La libreria deve essere scaricata (ad esempio sul Desktop) e inclusa nell'elenco delle librerie standard seguendo il percorso evidenziato in figura 6.

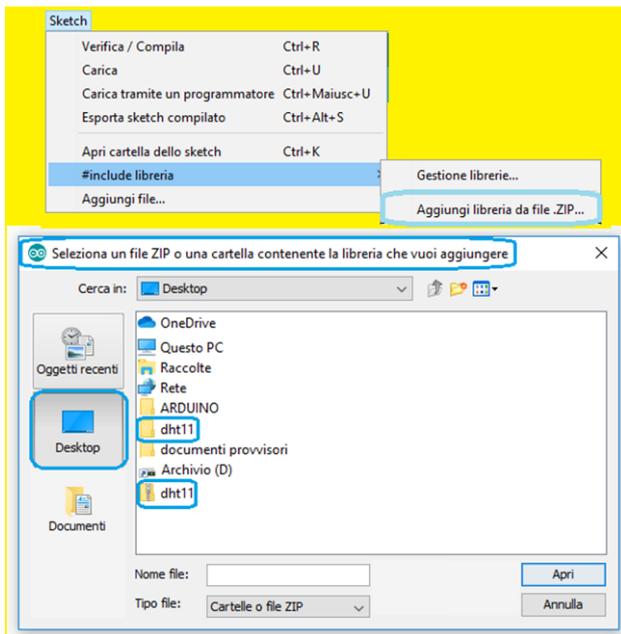


Figura 6

I dati provenienti dal sensore collegato ad una scheda Arduino possono essere utilizzati per scopi differenti.

Possono ad esempio essere visualizzati in diversi modi (sul monitor seriale, su display a 7 segmenti o su display a cristalli liquidi) o utilizzati per comandare degli attuatori (motori, ventole).

Il programma che segue, opportunamente commentato, acquisisce i dati provenienti dal sensore, visualizza sul monitor seriale i risultati ottenuti e comanda due ventole in relazione ai risultati acquisiti.

```
// il programma acquisisce i dati dal sensore DHT11 ad intervalli di 5 s
// i dati acquisiti vengono visualizzati sul monitor seriale
// in relazione ai dati acquisiti vengono attivate due ventole collegate ai pin 3 e 4
// quando la temperatura supera i 30 °C
// vengono attivate entrambe le ventole
// quando la temperatura è compresa tra 24 °C e 30 °C e l'umidità supera il 70 %
// viene attivata soltanto la ventola collegata al pin 3
#include <dht11.h> // si include la libreria dht11.h
dht11 DHT; // l'oggetto dht11 viene denominato DHT
int u,t; // u=umidità, t=temperatura
void setup(){
  Serial.begin(9600); // velocità di apertura della porta seriale
  pinMode (3,OUTPUT); // uscita ventola
  pinMode (4,OUTPUT); // uscita ventola
  // visualizzazione intestazione
  Serial.println();
  Serial.println("I dati vengono letti dal sensore DHT11 ad intervalli di 5 s");
  Serial.println("se t>30 °C si attivano entrambe le ventole");
  Serial.print("se 24 °C<t<30 °C e u>70 % si attiva soltanto la ventola collegata al pin 3");
  Serial.println();
  Serial.println();
}
void loop(){
```

```

DHT.read(2); // i dati vengono acquisiti dal pin 2
// visualizzazione dei dati sul monitor seriale
Serial.println("umidità (%)\t\ttemperatura (°C)");
u=DHT.humidity;
Serial.print(u); // visualizza il valore di umidità
Serial.print("\t\t");
t=DHT.temperature;
Serial.println(t); // visualizza il valore di temperatura
if (t>30){
Serial.println("stato ventola pin 3: attiva");
Serial.println("stato ventola pin 4: attiva");
digitalWrite(3,HIGH);
digitalWrite(4,HIGH);
}
if (u>70 && t>=24 && t<=30) {
Serial.println("stato ventola pin 3: attiva");
Serial.println("stato ventola pin 4: non attiva");
digitalWrite(3,HIGH);
digitalWrite(3,LOW);
}
if (t<24){
Serial.println("stato ventola pin 3: non attiva");
Serial.println("stato ventola pin 4: non attiva");
digitalWrite(3,LOW);
digitalWrite(4,LOW);
}
Serial.println();
delay(5000); // intervallo tra due letture successive
}

```

In figura 7 viene riprodotto un esempio di risultato dell'acquisizione che mostra i valori di umidità e temperatura unitamente allo stato di attivazione degli attuatori.

```

COM3 (Arduino/Genuino Uno)
I dati vengono letti dal sensore DHT11 ad intervalli di 5 s
se t>30 °C si attivano entrambe le ventole
se 24 °C<t<30 °C e u>70 % si attiva soltanto la ventola collegata al pin 3
umidità (%)          temperatura (°C)
76                   24
stato ventola pin 3: attiva
stato ventola pin 4: non attiva
umidità (%)          temperatura (°C)
95                   31
stato ventola pin 3: attiva
stato ventola pin 4: attiva

```

Figura 7