

LABORATORIO DIDATTICO 4 Misura di temperatura e umidità e rappresentazione grafica dei valori su smartphone

Lo scopo dell'esempio è misurare la temperatura e l'umidità mediante un sensore DHT22 e rappresentare i valori, mediante l'applicazione Android *"Bluetooth Electronics"*, su opportuni indicatori grafici.

Tale applicazione consente di far scambiare dati tra Arduino e uno smartphone Android, agendo su pannelli che possono essere personalizzati con pulsanti, slider, indicatori, pad, diagrammi temporali, ecc.

Alcune informazioni sull'applicazione Android *"Bluetooth Electronics"*:

- nella schermata iniziale (Figura 1a) sono rappresentati dei pannelli numerati, alcuni (demo) sono già impostati, altri sono vuoti e personalizzabili dall'utente;
- in alto a destra premere *"Connect"* e poi *"Bluetooth Classic"* per connettersi al modulo HC-XX;
- nella schermata successiva premere *"Discover"*, per individuare il modulo HC-XX acceso, selezionarlo e premere *"Pair"* (se richiesto digitare il pin 1234), quindi selezionarlo nella lista a destra e premere *"Connect"* e poi *"Done"*. A questo punto in alto sulla schermata iniziale viene segnalata la connessione e il lampeggio sul modulo HC-XX rallenta.
- Utilizzo di un pannello: selezionare il pannello 6 (*Temperature and Humidity Demo*) e premere *"Run"* per avviare la visualizzazione delle misure ricevute dal sensore DHT22: sullo schermo (Figura 1b) appaiono due indicatori che rappresentano i valori inviati da Arduino, sia numericamente che su scala graduata.

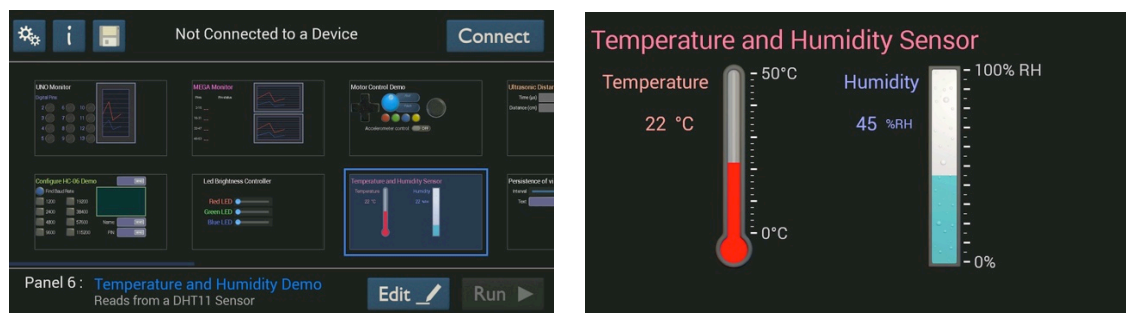


Figura 1 - Applicazione *"Bluetooth Electronics"*: a) schermata iniziale; b) pannello 6 per la misura di temperatura e umidità.

- Creazione o modifica di un pannello (Figura 2): una volta selezionato un pannello, premendo *"Edit"* è possibile modificarne la struttura e il funzionamento, oppure crearne uno nuovo. Nella finestra di editing sulla destra c'è la lista degli oggetti inseribili nel pannello; ad esempio, selezionando *"Indicators"* compaiono in basso tutti i possibili indicatori, ottici o acustici, che possono essere trascinati nel campo di lavoro per costruire il pannello. In basso a destra sono evidenziate le principali caratteristiche dell'oggetto selezionato; premendo *"Edit"* si possono modificare le caratteristiche dell'oggetto: fondo scala, numero delle divisioni, *Receive Character*, ecc.
- Comunicazione Arduino - Smartphone: ogni oggetto ha i suoi caratteri chiave per la comunicazione dei parametri. Ad esempio, per rappresentare il valore 40 sull'indicatore (*Gauge*) della temperatura in Figura 2, Arduino deve inviare la stringa **T40**, dove il carattere *'T'* (*Receive Character*) è modificabile nella finestra *"Edit"*. Per la scrittura delle istruzioni dello sketch, si vedano nella finestra *"Edit"* di ogni oggetto inserito nel pannello le relative regole di comunicazione.

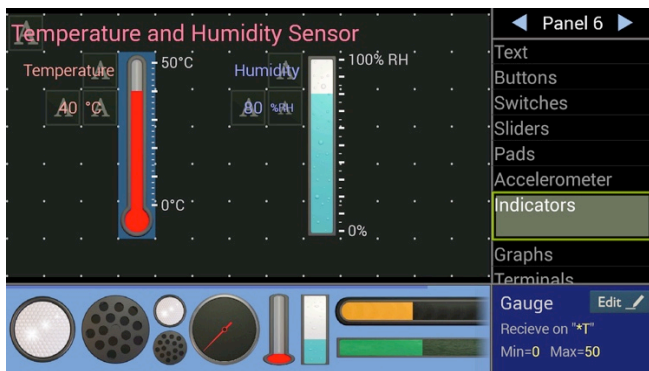


Figura 2 – Schermata di editing di un pannello.

Caratteristiche principali del sensore di temperatura e umidità DHT22 (**Figura 3**) :

- Campi di misura: Umidità relativa (RH) 20-90%, temperatura 0-50 °C.
- Precisione: $\pm 2\% \text{RH}$; $\pm 0,5\text{ }^{\circ}\text{C}$. Risoluzione: $0,1\% \text{RH}$; $0,1\text{ }^{\circ}\text{C}$.
- Alimentazione: 3,0 V – 5,5 V.
- Tempo di rilevazione medio: 2 s.
- Massima lunghezza del collegamento: 30 m.
- Struttura della stringa dati di 40 bit (5 byte) inviata sul pin DATA:
 - bit 1-16 (16 bit): 10 x valore dell'umidità (RH);
 - bit 17-32 (16 bit): 10 x valore della temperatura.
 - bit 33-40 (8 bit): checksum per la rivelazione di errori di trasmissione.
- Il pin DATA, che deve essere collegato a un resistore di pull-up da $4,7\text{ k}\Omega$, svolge funzione sia di input che di output:
 - *Input*: un livello BASSO di almeno 18 ms ricevuto da Arduino rappresenta la richiesta di trasmissione della stringa dati.
 - *Output*: la trasmissione dati del DHT22 inizia con un livello BASSO ($80\text{ }\mu\text{s}$) e uno ALTO ($80\text{ }\mu\text{s}$), seguiti dai 40 bit di dati. Ogni bit è costituito da un livello BASSO ($50\text{ }\mu\text{s}$) seguito da uno ALTO, la cui durata esprime il bit: $27\text{ }\mu\text{s}$ corrisponde allo 0 logico, $70\text{ }\mu\text{s}$ corrisponde all'1 logico.

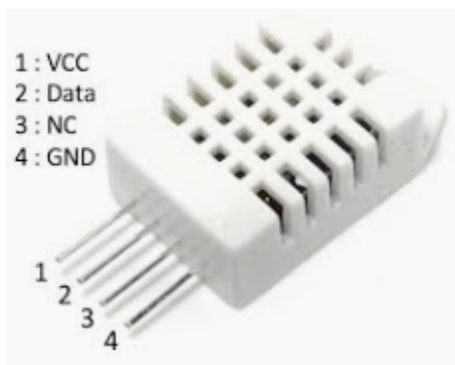


Figura 3 – Pin-out del sensore DHT22.

Il collegamento di Arduino con il sensore DHT22 e il modulo Bluetooth HC-XX è rappresentato in **Figura 4**.

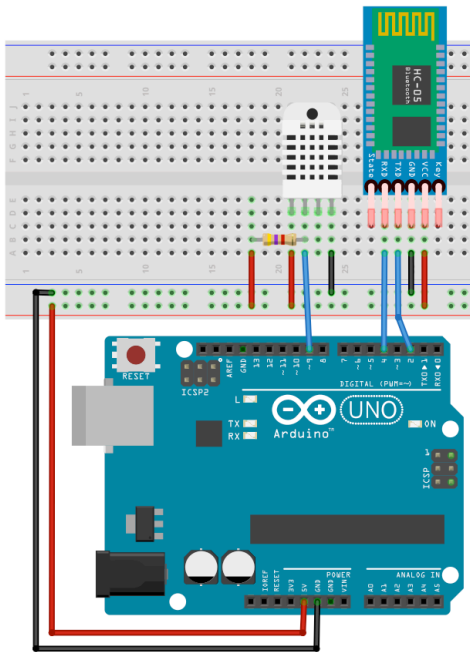


Figura 4 – Collegamento dei componenti.

Lo sketch di Arduino (con l'uso della libreria per il DHT22)

Lo sketch deve:

- gestire l'acquisizione dei dati di umidità e temperatura dal pin DATA del sensore DHT22: con l'impiego della libreria <DHT.h> sono sufficienti le istruzioni `dht.readHumidity()` e `dht.readTemperature()` per acquisire i dati dal sensore.
- Ritrasmettere i dati verso il Bluetooth: con i bit ricevuti dal sensore si costruiscono le stringhe per comunicare i valori all'applicazione Android *"Bluetooth Electronics"*.

```

/*
Misura di umidità e temperatura con il sensore DHT22.
Visualizzazione grafica sull'applicazione: Android "Bluetooth Electronics",
pannello: "Temperature and Humidity Demo".
Sul Serial Monitor vengono scritti i valori di umidità e temperatura misurati
dal DHT22 e trasmessi via Bluetooth.

Installa le librerie per il sensore DHT22:
Sketch --> #include libreria --> Gestione librerie --> cerca DHT -->
--> DHT sensor library --> seleziona la versione e installa
*/

#include <DHT.h>
#include <DHT_U.h>
#include <SoftwareSerial.h>
SoftwareSerial SerialBlue (2, 4); // comunicazione verso il modulo HC-XX
                                   // pin di Arduino: 2=RX ; 4=TX
int dataPin = 9;
DHT dht(dataPin, DHT22); // inizializza il sensore DHT22
unsigned int temp;        // temperatura in °C
unsigned int humidity;     // umidità relativa (%RH)

void setup() {

```

```

SerialBlue.begin(9600); // inizializza comunicazione via Bluetooth
Serial.begin(9600);      // inizializza Serial Monitor
dht.begin();             // inizializza comunicazione con il DHT22
}
void loop() {

  delay (2000); // pausa di 2 s prima di una nuova richiesta di misura

  humidity = dht.readHumidity(); // leggi l'umidità dal DHT22
  temp = dht.readTemperature(); // leggi la temperatura dal DHT22

  // trasmetti temperatura e umidità su Bluetooth in formato '*Txx*' e '*Hxx*'
  SerialBlue.print ("*T"+String(temp)+"*");
  SerialBlue.print ("*H"+String(humidity)+"*");

  // scrivi umidità e temperatura su Serial Monitor
  Serial.print("Umid: ");
  Serial.print(humidity);
  Serial.print(" %, Temp: ");
  Serial.print(temp);
  Serial.println(" Celsius");
}

```

Lo sketch senza l'uso della libreria per il DHT22

Per familiarizzare con la modalità di scambio dati del sensore DHT22, si ripropone lo sketch precedente senza fare uso di librerie specifiche.

Come descritto sopra, Arduino dopo aver fornito un impulso BASSO di 18 ms sul pin DATA del DHT22, riceve dallo stesso pin i 40 bit con i dati: il valore espresso dai bit 1-16 diviso per 10 rappresenta l'umidità (RH), mentre il valore espresso dai bit 17-32 diviso per 10 rappresenta la temperatura in °C.

Per poter verificare la corrispondenza dei bit letti dal sensore con i valori scritti sullo smartphone, alla fine dello sketch tutti i bit della stringa e i corrispondenti valori di temperatura e di umidità sono scritti sul Serial Monitor.

```

/*
Misura di umidità e temperatura con un sensore DHT22 e visualizzazione sull'applicazione
Android "Bluetooth Electronics", pannello "Temperature and Humidity Demo".
I dati di umidità e temperatura sono ricavati senza l'uso delle librerie per il DHT22,
analizzando i bit della stringa inviata dal sensore.
Sul Serial Monitor vengono scritti i bit inviati dal DHT22 e le corrispondenti stringhe
trasmesse via Bluetooth.
*/
#include <SoftwareSerial.h> // libreria comunicazione verso il modulo HC-XX
SoftwareSerial SerialBlue (2, 4); // collegato con i pin di Arduino: 2=RX ; 4=TX

int dataPin = 9;
boolean stringaRic[41]; // stringa ricevuta dal DHT22 (compreso bit di start)
unsigned int temp;      // temperature in °C
unsigned int humidity;  // umidità relativa (%RH)

```

```

void setup() {

  SerialBlue.begin(9600); // inizializza comunicazione via Bluetooth
  Serial.begin(9600);     // inizializza Serial Monitor
}

void loop() {

  delay (2000); // pausa di 2 s prima di una nuova richiesta di misura

  // porta basso il pin DATA per 18 ms per richiedere nuovi dati
  pinMode (dataPin, OUTPUT); // pin 9 OUTPUT (da Arduino a DHT22)
  digitalWrite (dataPin,LOW);
  delay(18);
  digitalWrite (dataPin,HIGH);

  pinMode (dataPin, INPUT); // riporta il pin 9 in modalità INPUT
                           // per ricevere i dati dal DHT22

  // leggi 41 bit dal DHT22 (il 1° impulso di start va poi ignorato)
  for (int i=0; i<=40; i++){
    stringaRic[i]=(pulseIn(dataPin, HIGH)>40); // pulseIn restituisce
                                                // la durata in µs dell'impulso a livello alto,
  }                                                // se è >40 µs il bit è un 1, altrimenti è 0

  // ricava il valore di umidità dai bit 1-16 dell'array stringaRic
  humidity=0;
  for (int i=1; i<=16; i++){ // leggi i 16 bit dell'umidità
    humidity=humidity<<1;    // shift SX di una posizione
                              // del valore binario di humidity
    if (stringaRic[i]) humidity |= 1; // bitwise compound OR: se l'iesimo bit
                                      // trasmesso è 1, questo viene sostituito
                                      // allo 0 nel LSB di humidity,
  }                                  // altrimenti rimane 0
  humidity=humidity/10; // per ottenere il valore reale si deve dividere
                       // per 10 quello ottenuto dalla stringa

  // ricava il valore di temperatura dai bit 17-32 dell'array stringaRic
  temp=0; // stesso procedimento di humidity
  for (int i=17; i<=32; i++){
    temp=temp<<1;
    if (stringaRic[i]) temp|=1;
  }
  temp=temp/10;

  // trasmetti i dati su Bluetooth in formato '*Txx*' e '*Hxx*'
  SerialBlue.print ("*T"+String(temp)+"*"); // trasmetti temperatura
  SerialBlue.print ("*H"+String(humidity)+"*"); // trasmetti umidità

  // scrivi su Serial Monitor i 40 bit della stringa e i valori trasmessi su Bluetooth
  for (int i=0; i<=40; i++){
    Serial.print (stringaRic[i]);
  }
}

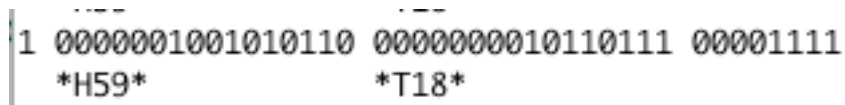
```

```

    if (i==0 || i==16 || i==32) Serial.print (" ");
}
Serial.println (" ");
Serial.print (" *H"+String(humidity)+"*");
Serial.print ("          ");
Serial.println ("*T"+String(temp)+"*");
}

```

La **Figura 5** mostra la sequenza di bit ricevuta dal sensore e visualizzata sul Serial Monitor, in corrispondenza di valori misurati: umidità = 59% , temperatura = 18 °C.



```

1 00000001001010110 0000000010110111 00001111
   *H59*              *T18*

```

Figura 5