

# DHT 11

Sensore di umidità e temperature

Progetto e caratteristiche

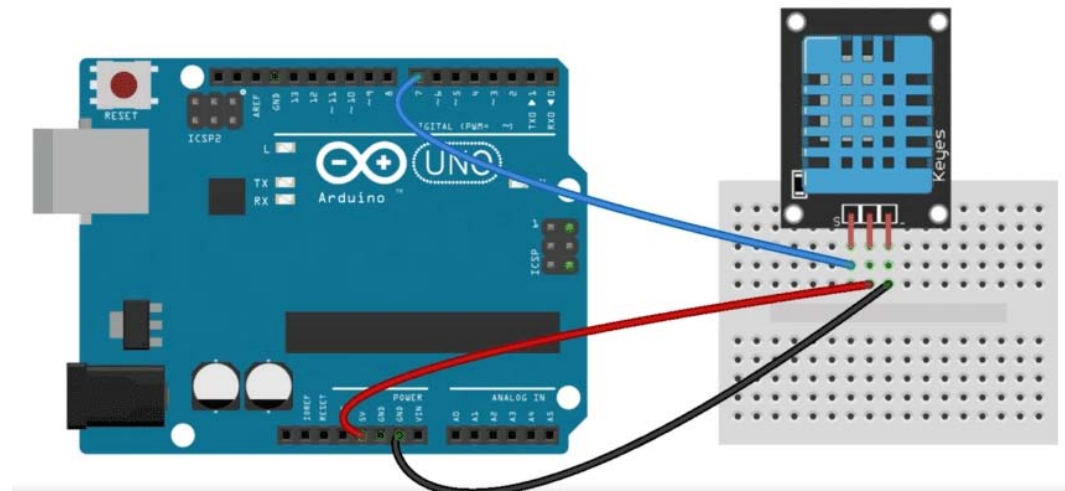
# Introduzione

- Il sensore DHT11 è un sensore di temperatura e umidità con uscita dei dati in formato digitale.
- I suoi elementi sensibili sono connessi con un processore 8-bit single-chip.
- Il sensore si presenta con un contenitore a 4 pin di uscita e compensato con un circuito che dà in uscita tre pin
- Le due grandezze in uscita vanno su un solo pin

# Caratteristiche

- La tensione di alimentazione è compresa tra 3-5.5V DC.
- Quando l'alimentazione viene fornita al sensore, non bisogna inviare alcuna istruzione al sensore per almeno un secondo per permettere al sensore di stabilizzarsi.
- Un condensatore da 100nF può essere inserito tra VDD e GND per il filtraggio dell'alimentazione
- **La Comunicazione del segnale avviene su un** single-bus utilizzato per la comunicazione tra con un MCU e sensore

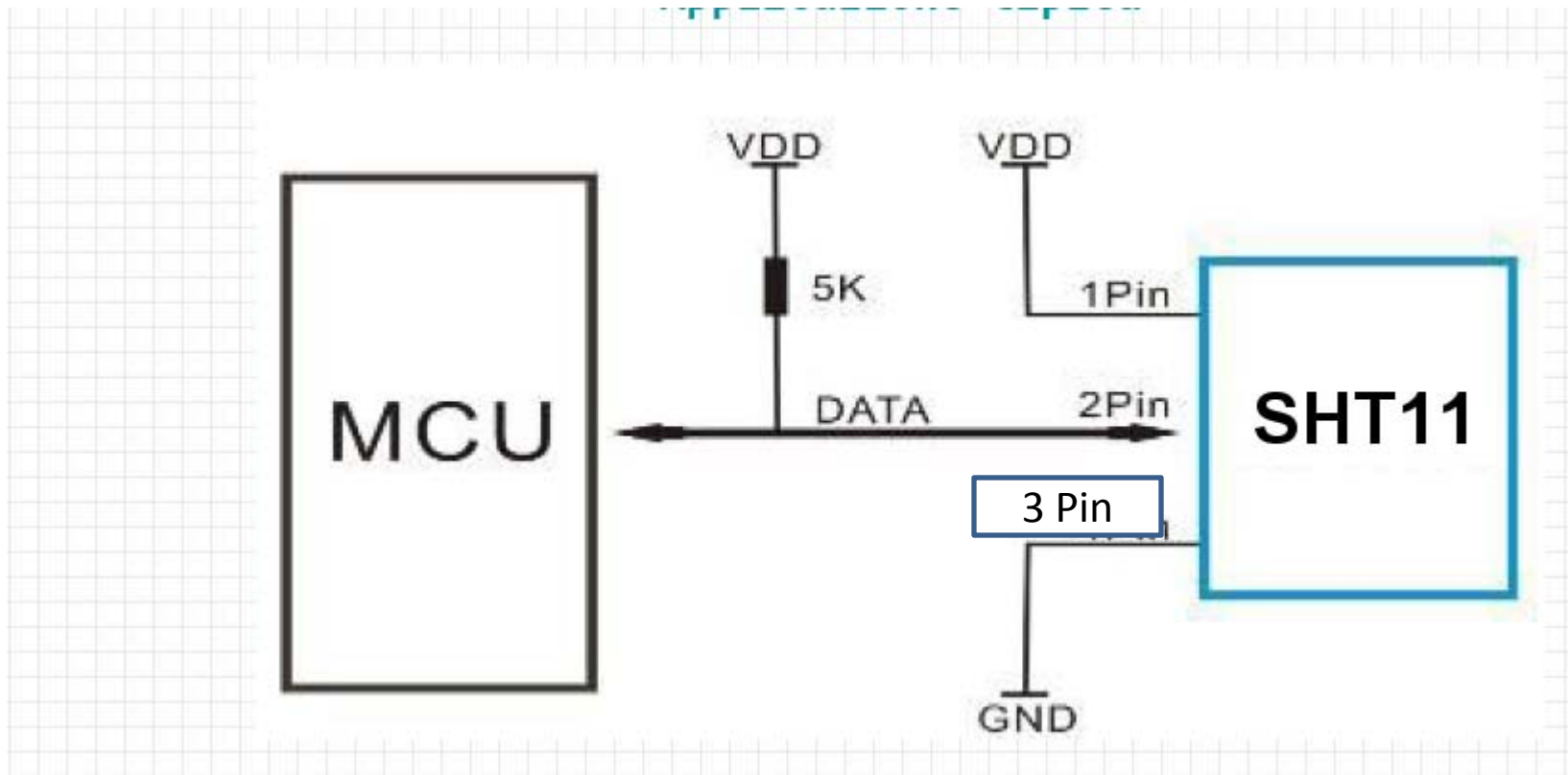
# Collegamenti



# Caratteristiche del sensore

<b>Modello</b>	DHT11
<b>Alimentazione</b>	3-5.5V DC
<b>Segnale di uscita</b>	digitale del segnale tramite single-bus
<b>Elemento sensibile</b>	Resistenza in Polimero
<b>Campo di misura umidità</b>	20-90% di umidità relativa, temperatura di 0-50 gradi Celsius
<b>Precisione</b>	umidità + -4% RH (Max + -5% di umidità relativa) temperatura +-2.0Celsius
<b>Risoluzione o la sensibilità</b>	umidità 1% di umidità relativa, temperatura 0.1 Celsius
<b>Ripetibilità umidità</b>	+ -1% di umidità relativa temperatura +-1Celsius
<b>Umidità isteresi</b>	+ -1% RH
<b>Stabilità a lungo termine</b>	+ -0.5% UR / anno
<b>Tempo di rilevazione</b>	medio: 2s
<b>Dimensioni</b>	12 * 15,5 * 5,5 millimetri

# Schema di funzionamento



# Progetto Arduino

- `#include <dht11.h>`
- `#define SENSOR_PIN 2`
- `void setup() {Serial.begin(9600);`
- `Serial.println("Humidity (%),\tTemperature (°C)");`
- `THSensor.begin(SENSOR_PIN);`
- `}`
- `void loop() {`
- `int result;`
- `result = THSensor.read(); //Read Data`
- `Serial.print(THSensor.humidity);`
- `Serial.print("\t\t");`
- `Serial.print(THSensor.temperature);`
- `Serial.println("\t\t");`
- `delay(1000);`
- `}`

# Programma Arduino

```
dht11_test$  
#include <dht11.h>  
#define SENSOR_PIN 2  
void setup() {Serial.begin(9600);  
  Serial.println("Humidity (%),\tTemperature (°C)");  
  THSensor.begin(SENSOR_PIN);  
}  
void loop() {  
  int result;  
  result = THSensor.read(); //Read Data  
  Serial.print(THSensor.humidity);  
  Serial.print("\t\t");  
  Serial.print(THSensor.temperature);  
  Serial.println("\t\t");  
  delay(1000);  
}
```

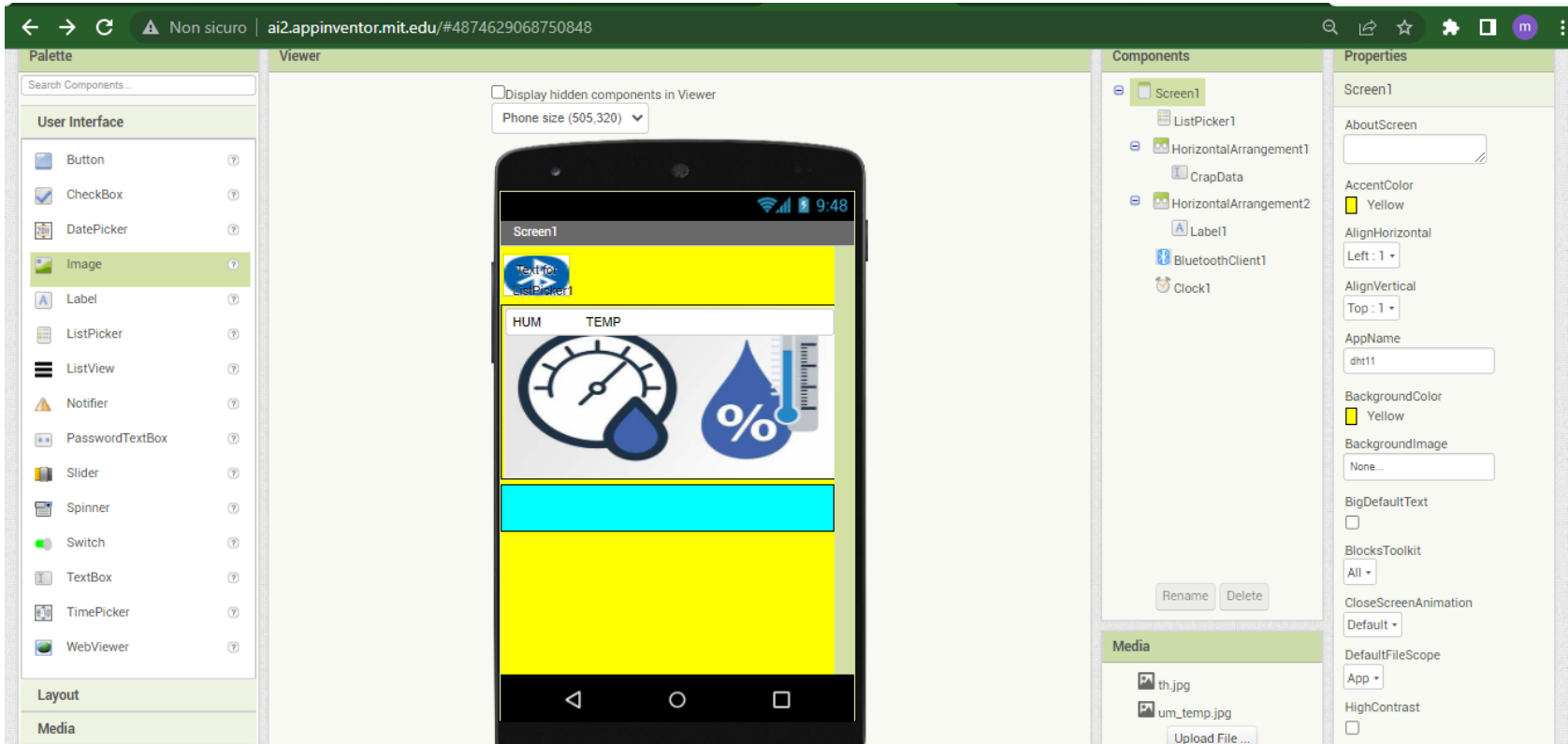
Trovata libreria non valida in C:\Program Files (x86)\Arduino\libraries\examples: Nessun header file (.h) trovato in C:\Pro  
Trovata libreria non valida in C:\Program Files (x86)\Arduino\libraries\ITEADLIB\_Arduino\_Nextion: Missing 'name' from libra

16 Arduino Uno su COM19





# Bluetooth



# Bluetooth

The screenshot displays a visual programming interface for a project named "dht11". The interface is divided into several sections:

- Top Bar:** Contains the project name "dht11", a "Screen1" dropdown, and buttons for "Add Screen...", "Remove Screen", and "Publish to Gallery". On the right, there are "Designer" and "Blocks" tabs.
- Blocks Panel (Left):** A sidebar with categories: Built-in, Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures, Screen1, ListPicker1, HorizontalArranger, CrapData, Label1, and BluetoothClient1. At the bottom of this panel are "Rename" and "Delete" buttons.
- Viewer (Center):** The main workspace containing a script of code blocks:
  - Initialize Global:** "initialize global name to" block with the value "0.0".
  - Before Picking:** A "when ListPicker1 . BeforePicking" block with a "do" block containing "get global name".
  - After Picking:** A "when ListPicker1 . AfterPicking" block with a "do" block containing "set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames".
  - Connect:** A "when ListPicker1 . Selection" block with a "do" block containing "set ListPicker1 . Selection to call BluetoothClient1 . Connect address ListPicker1 . Selection".
  - Timer:** A "when Clock1 . Timer" block with a "do" block containing:
    - An "if BluetoothClient1 . IsConnected" block.
    - A "then" block containing:
      - An "if call BluetoothClient1 . BytesAvailableToReceive > 0" block.
      - A "then" block containing:
        - "set global name to call BluetoothClient1 . ReceiveText numberOfBytes call BluetoothClient1 . BytesAvailableToReceive".
        - "set Label1 . Text to get global name".

- Bottom Left:** A "Show Warnings" button with a warning icon and a "0" count.
- Bottom Right:** A vertical toolbar with icons for zooming (eye, +, -) and deleting (trash).