

MIT App Inventor | Explore MIT x MIT App Inventor x

ai2.appinventor.mit.edu/#5348201707470848

Gmail Calendar Classroom Drive IIS J.C. Maxwell Tinkercad Tutte le novità della n... 19.04.2020 WeSchool | Login - W... >> Altri segnalibri

MIT APP INVENTOR Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English filippo.spadaro@maxwell.mi.it

LED_Bluetooth_DHT11_V3 Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Palette

Search Components...

User Interface

Layout

- HorizontalArrangement
- HorizontalScrollArrangement
- TableArrangement
- VerticalArrangement
- VerticalScrollArrangement

Media

Drawing and Animation

Maps

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

Extension

Viewer

Display hidden components in Viewer

Phone size (505,320)

Components

- Screen1
 - HorizontalArrangement2
 - ListPicker1
 - Button3
 - HorizontalArrangement1
 - Button1
 - Button2
 - VerticalArrangement1
 - Label1
 - Label2
 - BluetoothClient1
 - Clock1

Properties

Button2

BackgroundColor: Red

Enabled:

FontBold:

FontItalic:

FontSize: 14.0

FontTypeface: default

Height: Automatic...

Width: Fill parent...

Image: None...

Shape: default

ShowFeedback:

Text: OFF

TextAlignment: center : 1

TextColor: Default

Visible:

Non-visible components

- BluetoothClient1
- Clock1

Privacy Policy and Terms of Use

initialize global `dht11` to `0`

when `ListPicker1` .BeforePicking
do set `ListPicker1` .Elements to `BluetoothClient1` .AddressesAndNames

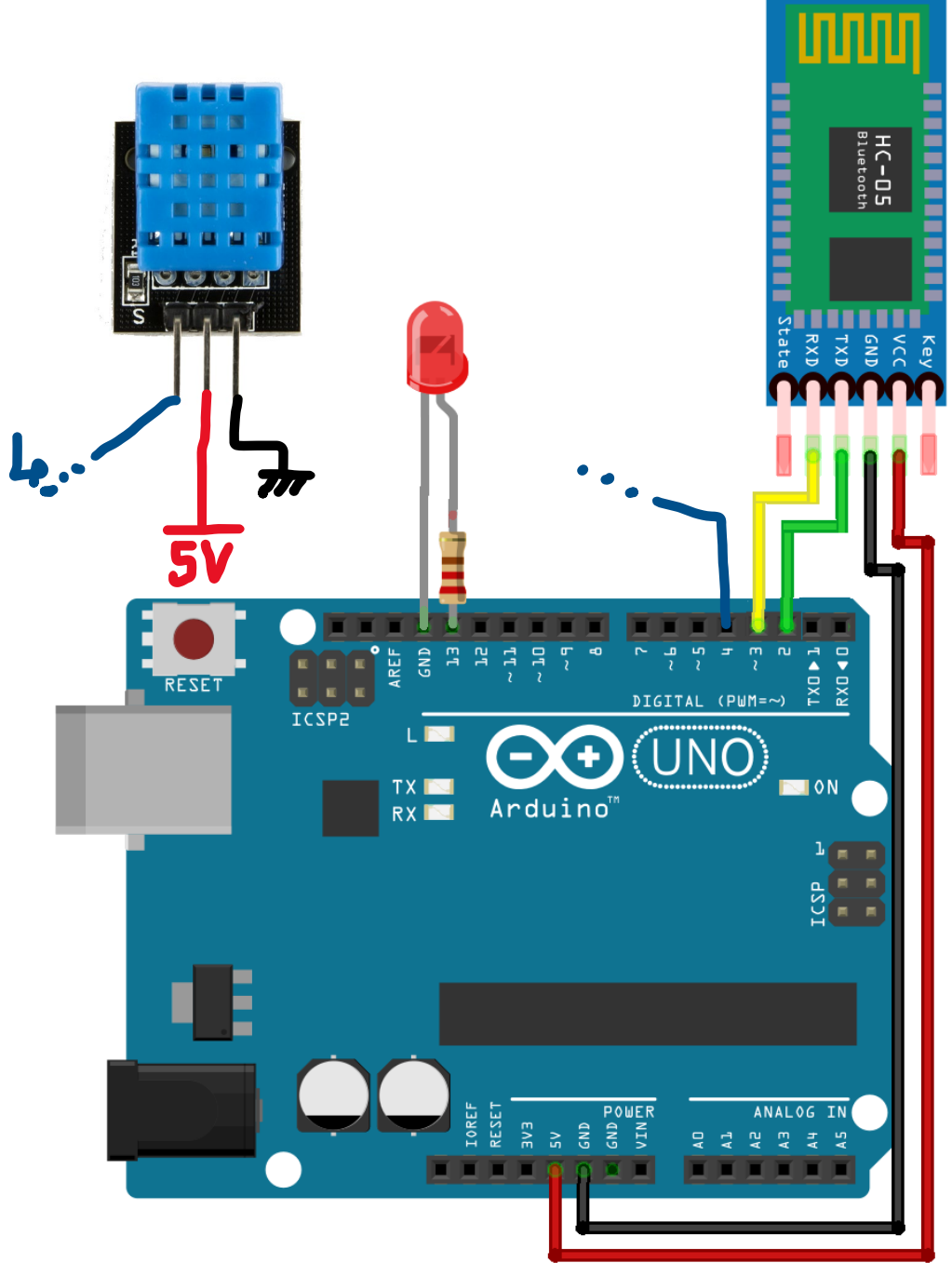
when `Button3` .Click
do call `BluetoothClient1` .Disconnect
set `ListPicker1` .Text to `"Connetti Bluetooth"`

when `ListPicker1` .AfterPicking
do set `ListPicker1` .Selection to call `BluetoothClient1` .Connect
address `ListPicker1` .Selection
set `ListPicker1` .Text to `"Connesso"`

when `Button1` .Click
do call `BluetoothClient1` .SendText
text `"1"`

when `Button2` .Click
do call `BluetoothClient1` .SendText
text `"0"`

when `Clock1` .Timer
do if `BluetoothClient1` .IsConnected
then if call `BluetoothClient1` .BytesAvailableToReceive > `0`
then set global `dht11` to call `BluetoothClient1` .ReceiveText
numberOfBytes call `BluetoothClient1` .BytesAvailableToReceive
if contains text get global `dht11`
piece `"-"`
then set `Label1` .Text to join `"Temperatura: "`
select list item list split text get global `dht11`
at `"-"`
index `1`
set `Label2` .Text to join `"Umidità: "`
select list item list split text get global `dht11`
at `"-"`
index `2`

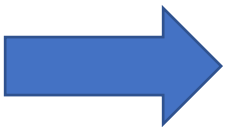


```
#include <SoftwareSerial.h>
#include <DHT.h>
#define intervallo 2000 // intervallo per il millis

DHT dht(4, DHT11);
SoftwareSerial Bluetooth(2, 3); // RX, TX

char Incoming_value = 0; // LED On = 1 / Off = 0
unsigned long tempo = 0; // tempo precedente per il millis
String temperatura, umidita; // valore acquisito dal DHT11 dopo casting

void setup()
{
  Serial.begin(9600); // seriale USB
  pinMode(13, OUTPUT); //LED Built In
  dht.begin();
  Bluetooth.begin(9600); // set the data rate for the SoftwareSerial port
}
```

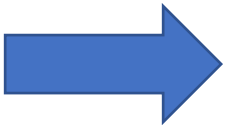


```
void loop()
{
  // gestione dati
  if ((millis()-tempo)>intervallo)
  {
    temperatura = String(dht.readTemperature());
    Bluetooth.print(temperatura); // invia al Bluetooth
    Serial.print("temperatura: ");
    Serial.print(temperatura);

    Bluetooth.print("-"); // invia al Bluetooth come discriminante

    umidita = String(dht.readHumidity());
    Bluetooth.print(umidita); // invia al Bluetooth
    Serial.print(" - umidita...: ");
    Serial.println(umidita);

    tempo = millis();
  }
}
```



```
// gestione LED On/Off
if(Bluetooth.available() > 0)
{
  Incoming_value = Bluetooth.read();
  Bluetooth.print("\n");
  if(Incoming_value == '1')
    digitalWrite(13, HIGH);
  else if(Incoming_value == '0')
    digitalWrite(13, LOW);
}
}
```