

Assuming that you are using the sketch version 3.3 here some tips to enable the temperature/humidity sensor:

1. opening the ANDRUINO_LIBS, you have to enable the DHT11 temperature/humidity sensor (THERMO_ADAFRUIT_DHT 1)

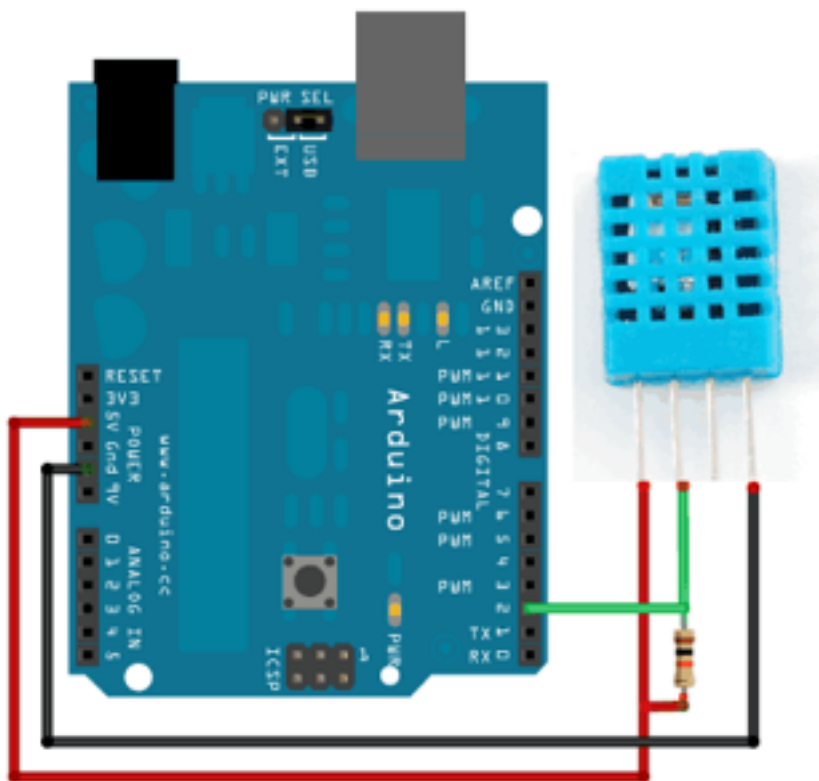
```
//ETHERNET SHIELD OR WIFI?  
#define ETHERNET_SHIELD 1  
#define WIFI_SHIELD 0  
  
//THERMOCOUPLE/HUMIDITY ADAFRUIT DHT11/22  
//Not validated on Arduino Uno and Due  
#define THERMO_ADAFRUIT_DHT 1  
  
//THERMOCOUPLE ADAFRUIT K MAX31855  
//Not validated on Arduino Uno and Due  
#define THERMO_ADAFRUIT_MAX31855 0  
  
//enable the ADC conversion for Irms calculation on adc0/adc1
```

2. Opening the Andruino_sketch the, at the beginning of the code the "if" will be enabled during the code compilation.

```
//Temperature/Humidity sensor  
//(uses ONE wire bus)  
#if THERMO_ADAFRUIT_DHT == 1 //check for details http://andruino-lib.com  
  #include <DHT.h>  
  #define DHTPIN 2 // PIN 2 (you can change it), serial  
  #define DHTTYPE DHT11 // can be DHT22 (AM2302), DHT21  
  DHT dht(DHTPIN, DHTTYPE);  
#endif
```

NOTE: By default the PIN 2 of Arduino is used to communicate with DHT11, but you can change it. This is the digital serial data communication between DHT11 and Arduino.

3. Use the below scheme to connect the DHT11. The resistor can be be 10Kohm.



4. In the main Sketch, disable the pinsetup of PIN2 as below (with THERMO_ADAFRUIT_DHT == 0):

```
#if (DEBUG_SERIAL == 0)
  pin_types.setupDig(ArduinoIO[0],0, INPUT,0);           //D0 is
  pin_types.setupDig(ArduinoIO[1],1, INPUT,0);           //D1 is
#endif

#if THERMO_ADAFRUIT_DHT == 0
  pin_types.setupDig(ArduinoIO[2],2, INPUT,0);           //can be us
#endif

pin_types.setupPWM(ArduinoIO[3],3, PWM,0);               //can be us
pin_types.setupDig(ArduinoIO[4],4, OUTPUT,0);           //can be us
```

5. To understand when the sensor is read, put some serial print in the main sketch:

```
//Adafruit DHT11/22/21 setup
#if THERMO_ADAFRUIT_DHT == 1
  if (check_adafruit2) { //read temper
    check_adafruit2 = false;
  }
  #if DEBUG_SERIAL == 1
    Serial.println(F("Reading DHT11/22"));
  #endif
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h))
    Arduino_User_var[0].value = 199; //mean
  else
    Arduino_User_var[0].value = h; //the
  if (isnan(t))
    Arduino_User_var[1].value = 199; //mean
  else
    Arduino_User_var[1].value = t; //the

  #if DEBUG_SERIAL == 1
    Serial.print(F("humidity: "));
    Serial.print(h);
    Serial.println(F("%"));
    Serial.print(F("temperature: "));
    Serial.print(t);
    Serial.println(F("C"));
  #endif
}
#endif
```

IMPORTANT: note that I have put as example the 2 data-output to Arduino_User_Var[0] and Arduino_User_Var[1] . These variables are also used by the IRMS current measurement, so if you are using the IRMS you have to change the index number.

If you have in ANDRUINO_DEFINES.h **#define POWER_CONSUMPTION_ENABLE 0**, no problem.

(So, at the end check if the variable used for temperature and humidity are used by other modules)

6. The DHT 11 is read every minute. I did it when "second_counter" reach 50 second, see below code in main sketch:

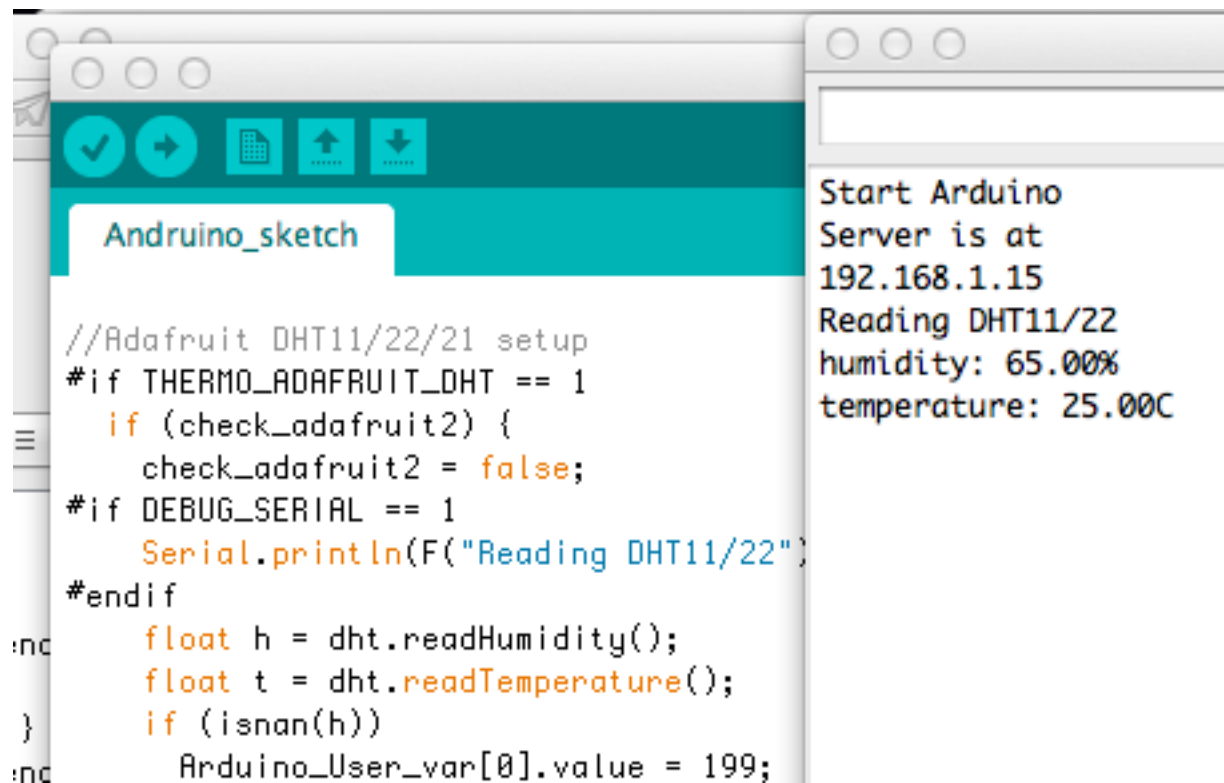
```
//Calculate seconds/minutes/hours
seconds_counter++;
if (seconds_counter %(CHECK_LIMITS EVERY) == 0)
    check_limits = true;
if (seconds_counter > 59) {
    seconds_counter = 0;
    minutes_counter++;
    if (minutes_counter %(CHECK_DDNS EVERY) == 0)
        check_ddns = true;
    if (minutes_counter > 59) {
        minutes_counter = 0;
        hours_counter++;
        if (hours_counter > 23) {
            hours_counter = 0;
            days_counter++;
        }
    }
}
if (seconds_counter == 30)
    check_adafruit1 = true;
if (seconds_counter == 50)
    check_adafruit2 = true;
```

NOTE: check_adafruit2 is set true, so the read of DHT11 will be done.

7. Enable the serial print (DEBUG_SERIAL 1) to see the temperature on datalog

```
//ENABLE SERIAL ONLY FOR DEBUG PRINTING
//WARNING, DEBUG SERIAL USE TOO RAM
#define DEBUG_SERIAL 1 //enable serial debug (use
#define DEBUG_SERIAL_JSON 0 //enable serial debug JSON
#define DEBUG_SERIAL_RAM 0 //enable serial debug FREE
#define DEBUG_SERIAL_ZIGBEE 0 //enable serial debug ZIGB
#define DEBUG_SERIAL_DDNS 0 //enable serial debug DDNS
#define DEBUG_SERIAL_MAIL 0 //enable serial debug MAIL
#define DEBUG_SERIAL_IRMS 0 //enable serial debug IRMS
```

8. Using Arduino IDE, and opening Instruments-->serial monitor you can see the Serial debug:



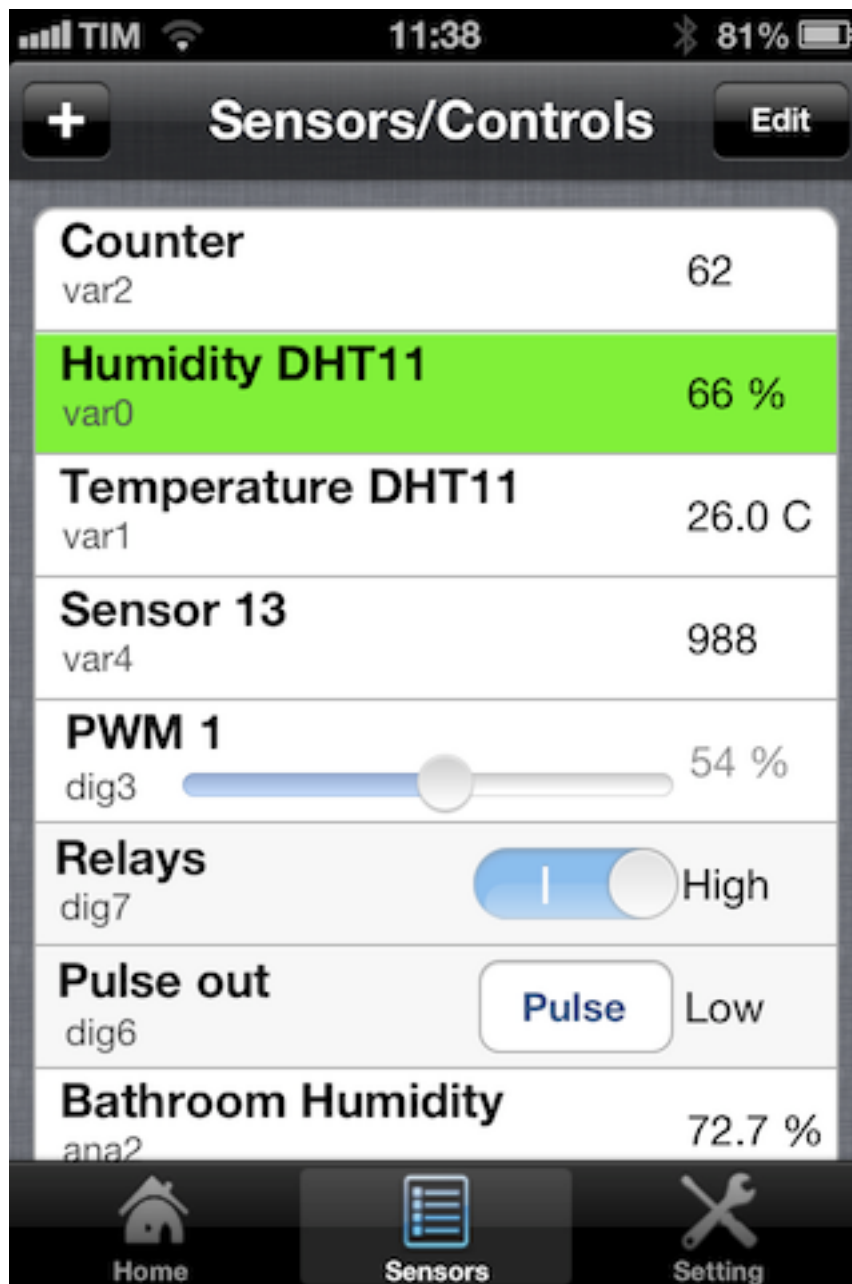
The image shows a screenshot of the Arduino IDE interface. On the left, a code editor window titled 'Andruino_sketch' displays the following C++ code:

```
//Adafruit DHT11/22/21 setup
#if THERMO_ADAFRUIT_DHT == 1
  if (check_adafruit2) {
    check_adafruit2 = false;
  }
#if DEBUG_SERIAL == 1
  Serial.println(F("Reading DHT11/22"));
#endif
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (!isnan(h))
    Arduino_User_var[0].value = 199;
```

On the right, the Serial Monitor window shows the output of the sketch:

```
Start Arduino
Server is at
192.168.1.15
Reading DHT11/22
humidity: 65.00%
temperature: 25.00C
```

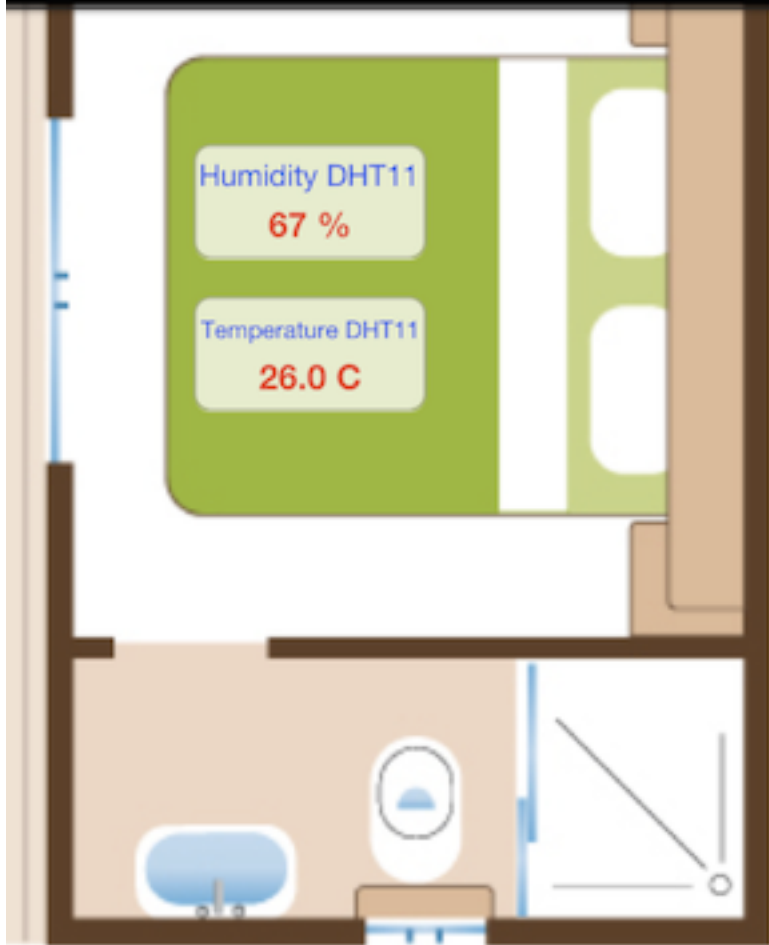
9. Using Andruino App, you can add a variable (ex: var0 and var1) to se the value above the home map




Cancel


Connected

Edit




Home


Sensors


Setting