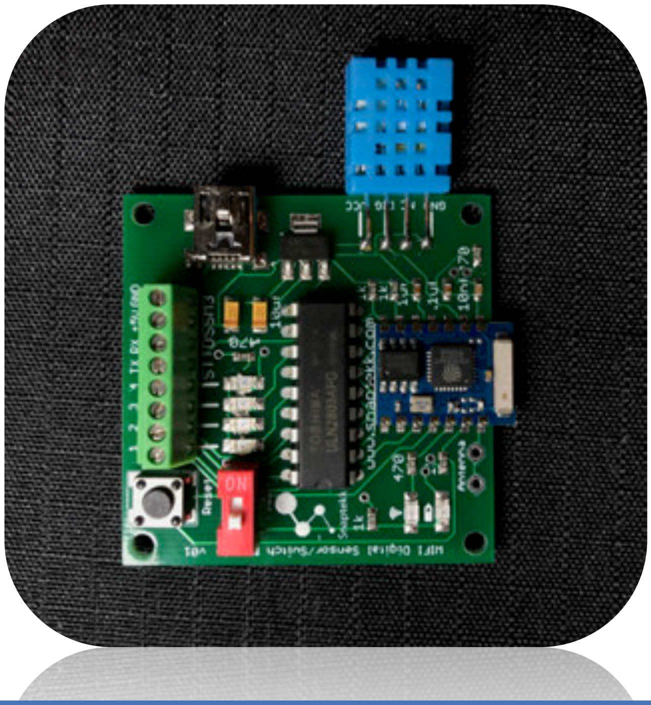
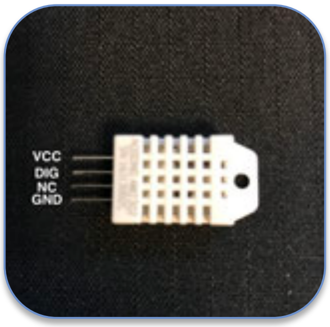
WiFi Temperature and Humidity Module
STTTH222 v02 – dweet.io and freeboard.io





Connecting the DHT22 Sensor

VCC: to VCC on module

DIG: to DIG on module

NC: No connection

GND: to GND on module

FUNCTIONAL DESCRIPTION

The Temperature and Humidity Module reads temperature and humidity values from a DHT11 and sends the data values in a message compatible with dweet.io and freeboard.io to make it convenient to display it. The message is sent to the server (dweet.io) at regular intervals (heartbeat). It can control 4 relays based on temperature and humidity limits.

One LED indicates power on; another LED indicates that the module has obtained an IP address and is connected to the WIFI network (color of the LEDs may change from model to model)

Ensure the orange dip switch is in the OFF position. This switch is only used for programming the board.

Connections:

Terminal pin 1: ON when temperature less than *Low Temperature Limit, otherwise OFF*
Terminal pin 2: ON when temperature higher than *High Temperature Limit, otherwise OFF*
Terminal pin 3: ON when humidity less than *Low Humidity Limit, otherwise OFF*
Terminal pin 4: ON when humidity higher than *High Humidity Limit, otherwise OFF*
TX: Serial transmit, only used for programming
RX: Serial receive, only used for programming
+5V: +5 volts DC power
GND: Ground from power supply
Mini USB Connector: Power the module from any Mini USB cable and power supply or computer.
VCC-DIG-NC-GND: Connect the DHT sensor, see picture above.

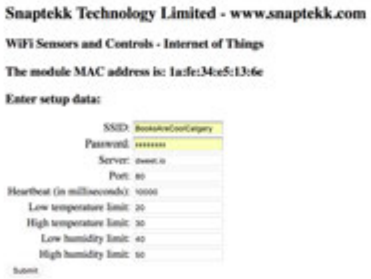
RELAY CONNECTIONS: You must connect the +5VDC and corresponding pin (1,2,3,4) to the relays, the module switches Ground.

WEB CONFIGURATION

Using a web browser

- Connect the 5V power making sure that polarity is correct.
- Press board reset button
- From your computer, check available wireless networks and connect to a network which access point name has the form STTTH222_xxxxxxx
- When prompted, enter the password for the STTTH222_xxxxxxx access point. The preset password is ‘espadmin’
- Wait until your computer successfully connects to STTTH222_xxxxxxx
- From your web browser and after station is connected enter http://192.168.4.1 and fill the information

  Where:
  SSID= your WIFI router SSID
  Password= your WIFI router password
  Server= the IP of the server to send data to (dweet.io)
  Port= the port of the server to send data to (80)
  Heartbeat= heartbeat frequency in milliseconds to send data to the server
  Low temperature limit= is the lowest temperature below which PIN1 will be switched ON
  High temperature limit= is the highest temperature beyond which PIN2 will be switched ON
  Low humidity limit= is the lowest humidity below which PIN3 will be switched ON
  High humidity limit= is the highest humidity beyond which PIN4 will be switched ON



Snaptekk Technology Limited - www.snaptekk.com

WiFi Sensors and Controls - Internet of Things

The module MAC address is: 1a:fe:34:e5:13:6e

Enter setup data:

SSID: BooksAreCoolCalgary
Password: ••••••••
Server: dweet.io
Port: 80
Heartbeat (in milliseconds): 10000
Low temperature limit: 20
High temperature limit: 30
Low humidity limit: 40
High humidity limit: 50

- Disregard any error messages in your browser and reset the board by pressing and releasing the reset button
- Once the module has connected to the WiFi network and obtained an IP address, the top LED will lit and the STTTH222_xxxxxxx access point name will change to STTTH222_C_xxxxxxx

OPERATING THE MODULE VIA DWEET.IO AND FREEBOARD.IO

Communicating with Dweet.io

Familiarize yourself with dweet.io by visiting their website at **dweet.io**.

You can communicate with the module via dweet.io by using POST and GET commands as follows:

To retrieve the last data values:

In dweet.io (play) use **GET /get/latest/dweet/for/{thing}** where

thing = STTTH222_ xxxxxxx

xxxxxxx is the unique module ID. Look in your available networks for a wifi signal named **STTTH222_** xxxxxxx

The module sends its data to dweet.io in the form:

POST /dweet/for/ STTTH222_XXXXXXXXX/?Temperature=23.5&Humidity=35.6

.

To send commands to the module:

You will send a command to the module whenever you want to modify the heartbeat, reset or reboot the module.

In dweet.io (play) use **POST /dweet/ for/{thing}** where

thing = params_ STTTH222_ xxxxxxx (NOTE THE params_ prefix)

xxxxxxx is the unique module ID. Look in your available networks for a wifi signal named **STTTH222_** xxxxxxx

content ={“command”:”the_command”,”value”=”the_value”}

Available commands (the_command) and values (the_value):

{“command”:”RD”,”value”=”0”} : Read data. The module will make a force read of the data and send it to the server

{“command”:”HB”,”value”=”xxxxx”} : change the heartbeat to xxxxx, where xxxxx is a value in milliseconds. Example: 5000 for 5 seconds heartbeat; 60000 for 1 minute heartbeat.

{“command”:”RS”,”value”=”0”} : Resets the module and wipe out SSID, Password, Server and Port configuration

{“command”:”RB”,”value”=”0”} : Reboots the module

{“command”:”LT”,”value”=”XX”} : Changes the low temperature limit to XX (integer)

{“command”:”HT”,”value”=”YY”} : Changes the high temperature limit to YY (integer)

{“command”:”LH”,”value”=”ZZ”} : Changes the low humidity limit to ZZ (integer)

{“command”:”HH”,”value”=VV”} : Changes the high humidity limit to VV (integer)

Displaying your data on freeboard.io

You can create beautiful dashboards with freeboard.io and monitor the data realtime. You can also send messages to the module.

To create a dashboard follow the freeboard.io instructions, but we will give you a quick intro here:

Go to freeboard.io and create an account.

Create a dashboard

Add a datasource of type dweet.io

Add a pane

Add a widget to the pane. Choose any of the widgets, usually you will choose Gauge, Text or Sparkline

Select the datasource (the one you just added above(

Select the data to display (Current)

Add as many widgets as needed binding to the datasource and the data to display.

The widgets will update realtime when the module sends data to dweet.io

To send commands to be module:

Add a widget of type “HTML”

On the HTML field, paste these HTML code:

```
<!doctype html>
 <html>
            <head>
            <title>Use a Link to send data to a server</title>
</head>
 <body>
<form action="http://dweet.io/dweet/for/params_STTTH222_xxxxxxx" target=_blank">
<select name="command" id="command_id">
  <option value="RD">Force Read</option>
  <option value="HB">Heartbeat</option>
  <option value="RB">Reboot</option>
  <option value="RS">Reset</option>
  <option value="LT">Low Temp Limit</option>
  <option value="HT">High Temp Limit</option>
  <option value="LH">Low Humidity Limit</option>
  <option value="HH">High Humidity Limit</option>
```

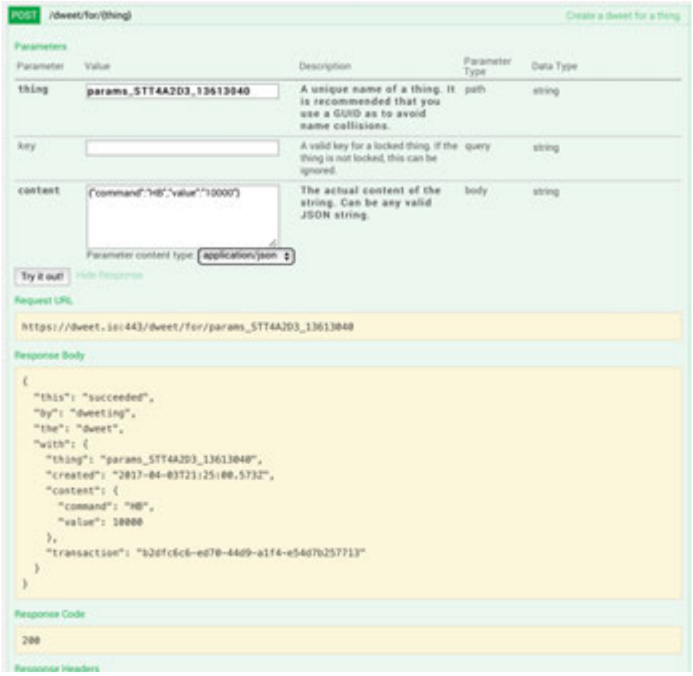
```
</select><br><br>
Value: <input type="text" name="value" value=""><br><br>
<input type="submit" value="Submit">
</form>
 </body>
 </html>
```

In *<form action="http://dweet.io/dweet/for/params_STTTH222_xxxxxxx" target=_blank">* change *xxxxxxx* for the module unique ID

NOTE THAT ANY COMMAND SENT TO THE MODULE WILL TAKE EFFECT ONLY WHEN THE HEARTBEAT CYCLE IS COMPLETED AND THAT ONLY ONE COMMAND (THE LAST ONE) IS EXCECUTED ON EACH HEARTBEAT CYCLE.

When sending a command, a new browser tab will be created with the server response, make sure you remove this tab every time a command is completed.

Examples of dweet.io messaging and freeboard.io dashboards below.



POST /dweet/for/{thing} Create a dweet for a thing

Parameters

Parameter	Value	Description	Parameter Type	Data Type
thing	params_STT4A2D3_13613040	A unique name of a thing. It is recommended that you use a GUID as to avoid name collisions.	path	string
key		A valid key for a locked thing. If the thing is not locked, this can be ignored.	query	string
content	{"command":"HB","value":"10000"} Parameter content type: application/json	The actual content of the string. Can be any valid JSON string.	body	string

Request URL

```
https://dweet.io:443/dweet/for/params_STT4A2D3_13613040
```

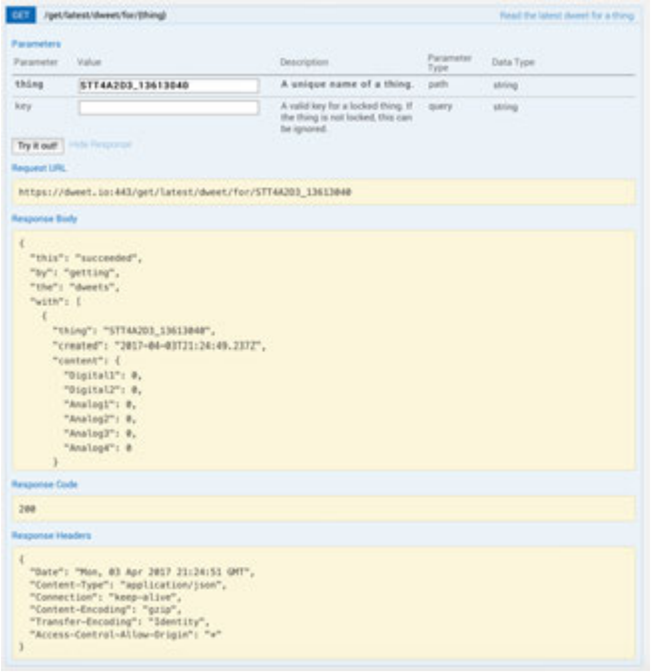
Response Body

```
{
  "this": "succeeded",
  "by": "dweeting",
  "the": "dweet",
  "with": {
    "thing": "params_STT4A2D3_13613040",
    "created": "2017-04-03T21:25:00.573Z",
    "content": {
      "command": "HB",
      "value": 10000
    },
    "transaction": "b2dfc6c6-ed70-44d9-a1f4-e54d7b257713"
  }
}
```

Response Code

```
200
```

Response Headers



GET /get/latest/dweet/for/{thing} Read the latest dweet for a thing

Parameters

Parameter	Value	Description	Parameter Type	Data Type
thing	STT4A2D3_13613040	A unique name of a thing.	path	string
key		A valid key for a locked thing. If the thing is not locked, this can be ignored.	query	string

Request URL

```
https://dweet.io:443/get/latest/dweet/for/STT4A2D3_13613040
```

Response Body

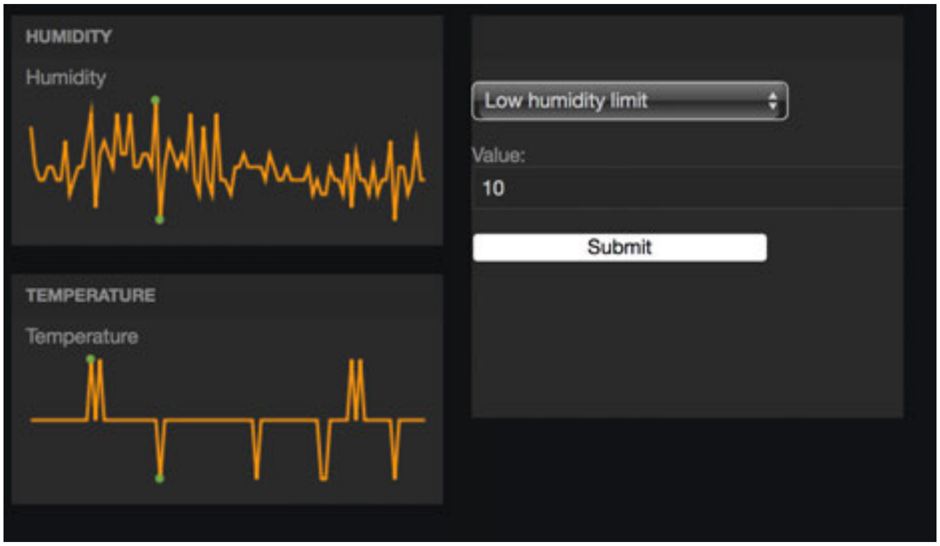
```
{
  "this": "succeeded",
  "by": "getting",
  "the": "dweets",
  "with": [
    {
      "thing": "STT4A2D3_13613040",
      "created": "2017-04-03T21:24:49.237Z",
      "content": {
        "Digital1": 0,
        "Digital2": 0,
        "Analog1": 0,
        "Analog2": 0,
        "Analog3": 0,
        "Analog4": 0
      }
```

Response Code

```
200
```

Response Headers

```
{
  "Date": "Mon, 03 Apr 2017 21:24:51 GMT",
  "Content-Type": "application/json",
  "Connection": "keep-alive",
  "Content-Encoding": "gzip",
  "Transfer-Encoding": "Identity",
  "Access-Control-Allow-Origin": "*"
}
```



Low humidity limit

Value: 10