

WiFi Integrated Analog/Digital Sensor/Switch Module Model STTISSM₃ – dweet.io and freeboard.io v03

The **WiFi Integrated Analog/Digital Module** features 4 configurable inputs/outputs, a fixed digital input and an analog input. The digital outputs can be used to control relays. It uses dweet.io and freeboard.io compatible formatting. It also features remote reset, reboot and OTA software update. Reset and clear settings button.

Compatible with **dweet.io** and **freeboard.io**

Fully configurable via web browser:

- SSID of WIFI router and Password
- Heartbeat configurable for data reporting to dweet.io server
- Server and port
- Minimum analog data value
- Maximum analog data value
- Data sent when digital changes or upon request

Specifications:

5VDC power input
80 ma transmitting/receiving
5ma stand by
4 digital inputs/outputs
1 digital input
1 analog input (10 bit, 0-1 Volt)
dweet.io and freeboard.io data formatted



Module layout may differ from picture

FUNCTIONAL DESCRIPTION

The Integrated Analog/Digital Sensor/Switch module has a dedicated analog input, a dedicated digital input and 4 digital pins that can be configured as inputs or as outputs via a dip switch and remote command. When selected as outputs, the state of the output (on/off) can be controlled via a POST message to the board via dweet.io or freeboard.io and the pins can drive 5Volt relays.

Input analog data limits are 0VDC to 1VDC, and this maps into output values of 0 to 1024 (10 bits)

Input digital data limits are zero, for a digital zero, and 3.3VDC for a digital 1

The module sends the data to the server and port specified at the heartbeat rate and when the digital input changes.

The LEDs are active when the corresponding pins are selected as output.

To select pins 1 to 4 as input or output, just flip the switch to the corresponding position (IN or OUT), and send the corresponding command to the module (IO10 for pin 1 output mode, IO11 for pin 1 input mode, see below for full description)

FLASHING THE MODULE

- Connect the board RX and TX to a serial adaptor
- Turn the Flash switch into the on position
- Press the reset button
- Start the flash program
- If flashing fails, press reset button one or two times and try again

WEB CONFIGURATION

Using a web browser

- Connect the 5V power
- Press board reset button
- From your computer, check available wireless networks and connect to a network which access point name has the form STTISSM3_XXXXXXX
- When prompted, enter the password for the STTISSM3_XXXXXXX access point. The preset password is 'espadmin'
- Wait until your computer successfully connects to STTISSM3_XXXXXXX
- From your web browser and after station is connected enter the following address, then respond with the desired parameters, including your WIFI network SSID and password, server IP address and port to send data to:

192.168.4.1

or enter the full address and command:

or **http://192.168.4.1/? SSID=Your_SSID&Password=password&Server=dweet.io&Port=80& Heartbeat=10000**

Where:

SSID= your WIFI router SSID

Password= your WIFI router password

Server= dweet.io (you could send the data to another server, just enter the IP address here)

Port= 80

Heartbeat= heartbeat frequency in milliseconds to send data to the server

- Reset the board
- Once the module has connected to the wifi network and obtained an IP address the STTISSM3_XXXXXXX access point name will change to STTISSM3_C_XXXXXXX (check your available WiFi networks)

NOTE: If your computer fails to connect to the module access point, press the reset button and the module will change the WiFi channel that it is using, wait 30 seconds to a minute for your computer to refresh the available WiFi networks and try again. You might have to do this several times if there are a lot of networks present.

OPERATING THE MODULE VIA DWEET.IO AND FREEBOARD.IO

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 = STTISSM3_XXXXXXX

XXXXXXX is the unique module ID. Look in your available networks for a wifi signal named STTISSM3_XXXXXXX

The module will respond with a message of the form:

POST /dweet/for/STTISSM3_XXXXXXXXX/?IO1=0&IO0=1&IO3=0&IOA=754&IO4=0&IO2=0

Where

IO0: dedicated digital input

IO1: configurable digital input/output 1

IO2: configurable digital input/output 2

IO3: configurable digital input/output 3

IO4: configurable digital input/output 4

IOA: analog input

To send commands to the module:

You will send a command to the module whenever you want to activate one of the switches, modify a parameter, reset or reboot the module.

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

thing = params_STTISSM3_xxxxxxx (NOTE THE params_ prefix)

xxxxxxx is the unique module ID. Look in your available networks for a wifi signal named STTISSM3_xxxxxxx

content ={"command":"the_command","value":"the_value"}

Available commands (the_command) and values (the_value):

{"command":"RD","value":"o"} : 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":"o"} : Resets the module and wipe out SSID, Password, Server and Port configuration

{"command":"RB","value":"xxxxx"} : Reboots the module

{"command":"IO1","value":"o"} : sets IO1 to output mode

{"command":"IO1","value":"1"} : sets IO1 to input mode

{"command":"IO2","value":"o"} : sets IO2 to output mode

{"command":"IO2","value":"1"} : sets IO2 to input mode

{"command":"IO3","value":"o"} : sets IO3 to output mode

{"command":"IO3","value":"1"} : sets IO3 to input mode

{"command":"IO4","value":"o"} : sets IO4 to output mode

{"command":"IO4","value":"1"} : sets IO4 to input mode

{"command":"SW1","value":"o"} : Switches IO1 to o (OFF). Only active if IO1 is set to output

{"command":"SW1","value":"1"} : Switches IO1 to o (ON). Only active if IO1 is set to output

{"command":"SW2","value":"o"} : Switches IO2 to o (OFF). Only active if IO2 is set to output

{"command":"SW2","value":"1"} : Switches IO2 to o (ON). Only active if IO2 is set to output

{"command":"SW3","value":"o"} : Switches IO3 to o (OFF). Only active if IO3 is set to output

{"command":"SW3","value":"1"} : Switches IO3 to o (ON). Only active if IO3 is set to output

{"command":"SW4","value":"o"} : Switches IO4 to o (OFF). Only active if IO4 is set to output

{"command":"SW4","value":"1"} : Switches IO4 to o (ON). Only active if IO4 is set to output

Example of dweet.io POST for modifying a parameter activating the switches and GET for retrieving data values:

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

Parameter	Value	Description	Parameter Type	Data Type
thing	params_STT15SM3_1034994	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	<code>{ "command": "RS", "value": 1 }</code>	The actual content of the string. Can be any valid JSON string.	body	string

Parameter content type: **application/json**

Try it out! Hide Response

Request URL

```
https://dweet.io:443/dweet/for/params_STT15SM3_1034994
```

Response Body

```
{
  "this": "succeeded",
  "by": "dweeting",
  "the": "dweet",
  "with": {
    "thing": "params_STT15SM3_1034994",
    "created": "2017-04-02T20:16:57.779Z",
    "content": {
      "command": "RS",
      "value": 1
    },
    "transaction": "a9fef4e7-155b-416c-88b7-185e798f0492"
  }
}
```

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

Parameter	Value	Description	Parameter Type	Data Type
thing	STT15SM3_1034994	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

Try it out! Hide Response

Request URL

```
https://dweet.io:443/get/latest/dweet/for/STT15SM3_1034994
```

Response Body

```
{
  "this": "succeeded",
  "by": "getting",
  "the": "dweets",
  "with": [
    {
      "thing": "STT15SM3_1034994",
      "created": "2017-04-02T19:43:30.738Z",
      "content": {
        "IO0": 0,
        "IO1": 1,
        "IO2": 1,
        "IO3": 0,
        "IO4": 1
      }
    }
  ]
}
```

Response Code

```
200
```

Response Headers

```
{
  "Date": "Sun, 02 Apr 2017 19:43:36 GMT",
  "Content-Type": "application/json",
  "Connection": "keep-alive",
  "Content-Encoding": "gzip",
  "Transfer-Encoding": "Identity",
  "Access-Control-Allow-Origin": "*"
}
```

freeboard.io

You can create beautiful dashboard 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, Indicator Light or Sparkline

Select the datasource (the one you just added above)

Select the data to display (IO₀, IO₁, IO₂, etc)

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_STTISMM3_xxxxxxx" target=_blank">
<select name="command" id="command_id">
  <option value="IO1">IO 1</option>
  <option value="IO2">IO 2</option>
  <option value="IO3">IO 3</option>
  <option value="IO4">IO 4</option>
  <option value="SW1">Switch 1</option>
  <option value="SW2"> Switch 2</option>
  <option value="SW3"> Switch 3</option>
  <option value="SW4"> Switch 4</option>
  <option value="RD">Force Read</option>
  <option value="HB">Heartbeat</option>
  <option value="RB">Reboot</option>
  <option value="RS">Reset</option>
  <option value="UD">Update Software</option>
</select><br><br>
Value: <input type="text" name="value" value=""><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>

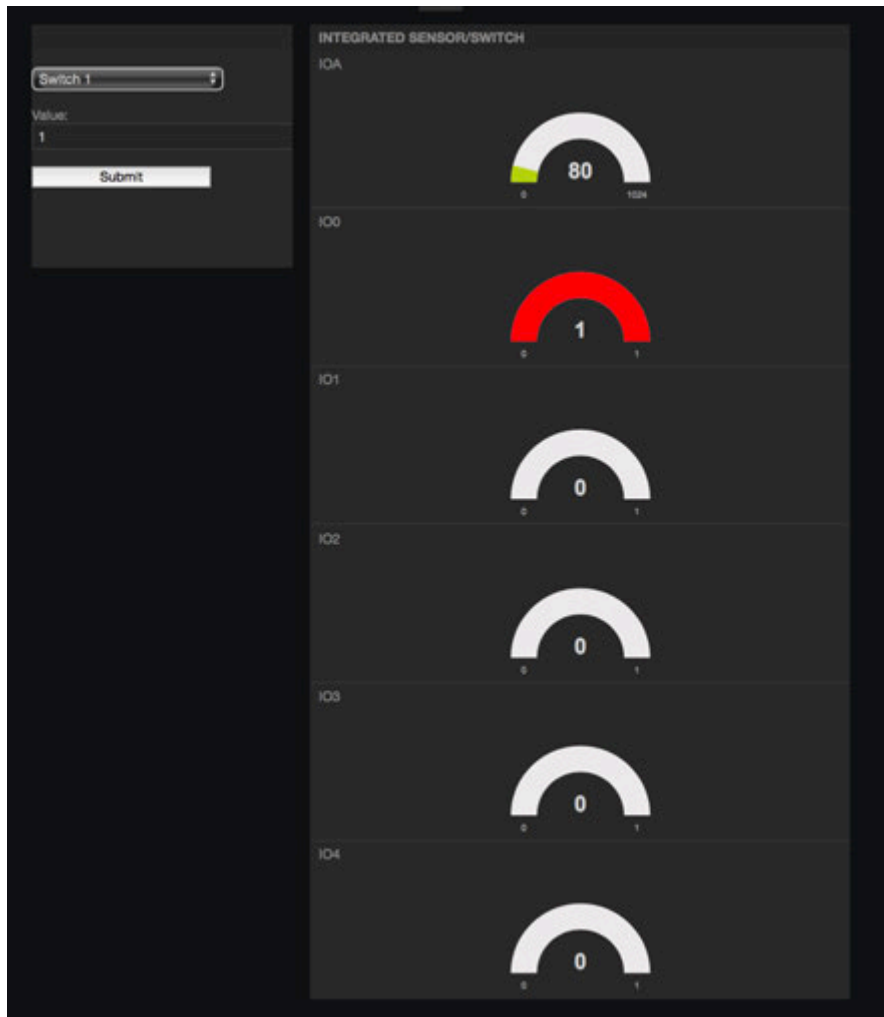
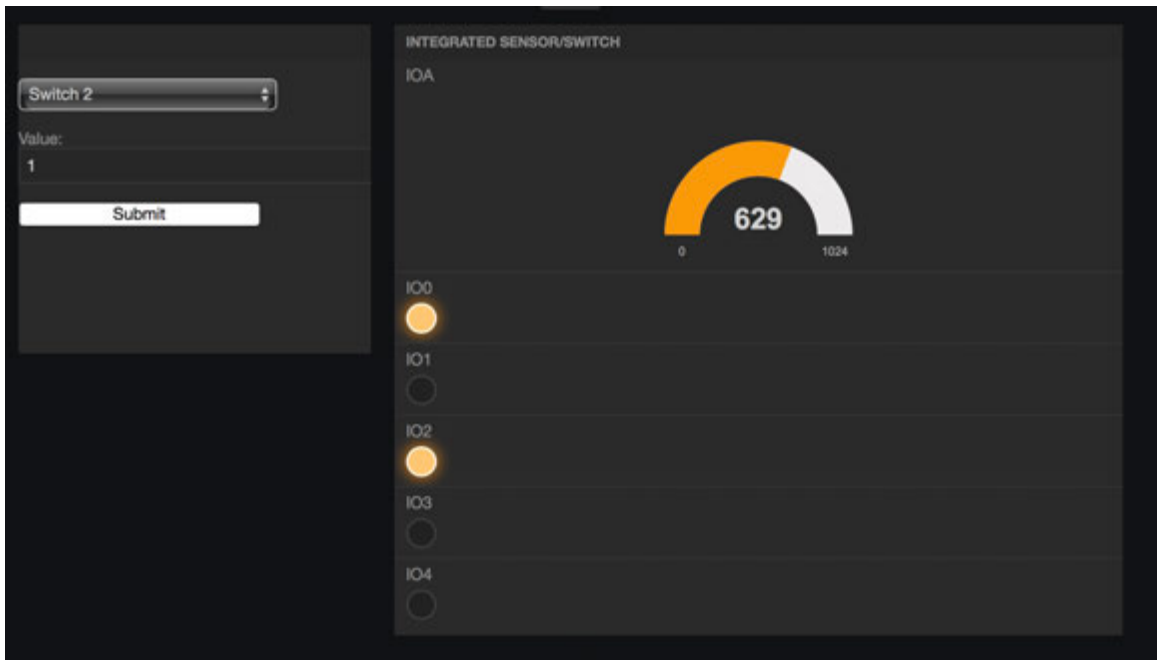
```

In `<form action="http://dweet.io/dweet/for/params_STTISMM3_xxxxxxx" target=_blank">` change xxxxxxx for the module unique ID

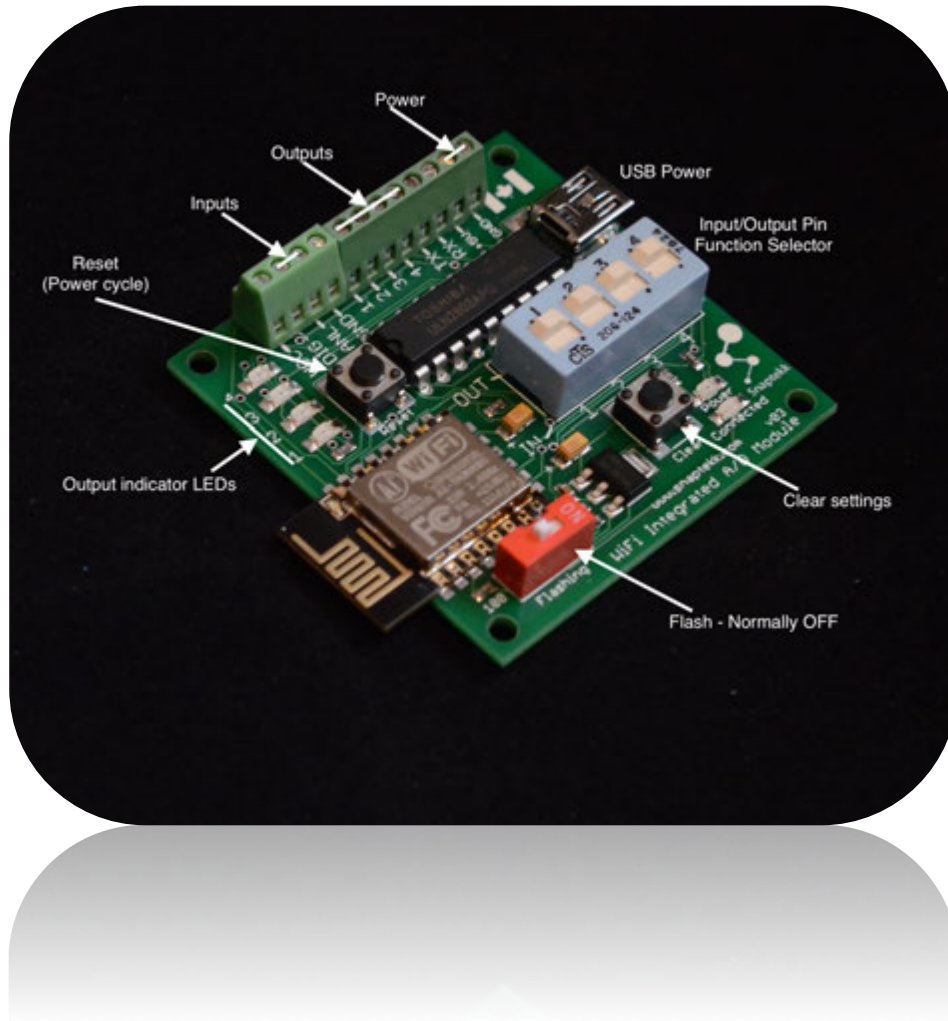
NOTE THAT ANY COMMAND SENT TO THE MODULE WILL TAKE EFFECT ONLY WHEN THE HEARTBEAT CYCLE IS COMPLETED AND THAT ONLY ONE COMMAND (THE LAST ONE) IS EXECUCUTED ON EACH HEARTBEAT CYCLE.

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 freeboard.io dashboards on the next page.



WIRING THE MODULE



DWEET.IO AND FREEBOARD.IO INTEGRATION

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 = STTISSM3_xxxxxxx

xxxxxxx is the unique module ID. Look in your available networks for a wifi signal named STTISSM3_xxxxxxx

The module will respond with a message of the form:

```
POST /dweet/for/STTISSM3_XXXXXXXXX/?IO1=0&IO0=1&IO3=0&IOA=754&IO4=0&IO2=0
```

To send commands to the module:

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

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

thing = params_STTISSM3_xxxxxxx

xxxxxxx is the unique module ID. Look in your available networks for a wifi signal named STTISSM3_xxxxxxx

The module will modify only one parameter per heartbeat cycle. If you want to modify many parameters, you need to carefully make sure the heartbeat cycle is completed before sending another command. Shorten the heartbeat cycle if this is inconvenient.

Available commands