



Arduino Lesson ()

Email alert for DHT11 data with Arduino WeMos D1 ESP8266 & Blynk app

Name: _____ () Class: _____ Date: _____

Objectives: At the end of this lesson, you would be able to

1. Connect a Arduino **WeMos D1** Wifi UNO ESP8266 to receive readings from DHT11
2. Write a sketch to use ESP8266 to send an email alert with Blynk app on a smartphone when temperature exceeds a certain value

Apparatus:

- 01 Arduino **WeMos D1 Wifi UNO ESP8266**
- 01 DHT11 temperature & humidity sensor module
- 01 *USB micro B cable (used for typical powerbank)
- 01 mini-breadboard
- 03 male-male jumper wires
- smartphone (with **Blynk** app downloaded)

USB micro B cable



Note:

- WEMOS D1 is a [WIFI development board](#) based on ESP8266 12E, and is built resembling Arduino UNO.
- It turns the very popular ESP8266 wireless (WiFi) module into a [fully fledged development board](#).

Assignments 1 & 2 below are similar to those in **Arduino Lesson (2020) for WeMos D1**

- “**Use of DHT 11 with Arduino WeMos D1 Wifi UNO ESP8266 & display of data on Blynk app**”
- Hence, only extracts are shown.

Assignment 1: Install the Arduino WeMos board on Arduino IDE

1. Connect the Arduino WeMos board to the laptop with USB micro B cable and check for connectivity:
 - Select **Tools** → **Board:** scroll down to select “**WeMos D1 R1**”
 - and select ***Port:** “COM#”

*Note:

- Must use proper **USB micro B cable**. Those typically used for powerbank **often don't have data wires** (data lines) (**they are charge only**) such that the **Port** may not be detected.
See <https://superuser.com/questions/1260407/are-all-usb-3-micro-b-cables-functionally-the-same>

Assignment 2: Set up DHT11 circuit with Arduino WeMos board

2. Insert the DHT11 sensor module onto the breadboard. Set up the DHT11 circuit using the diagram shown below.

With the sensor facing you, pins 1 to 3, from left to right:

DHT11 → **Arduino WeMos board**

Pin 1 (+) → 5 V

Pin 2 (out) → digital pin 4 (**D4**)

Pin 3 (-) → GND (ground)

[Pin number may vary according to DHT11 model.]

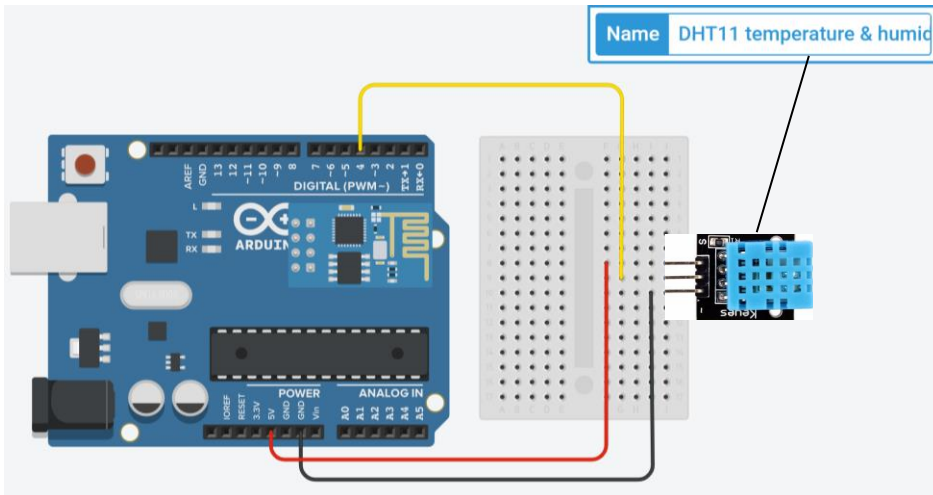


Diagram drawn in tinkercad.com (circuits) (DHT11 module added separately)

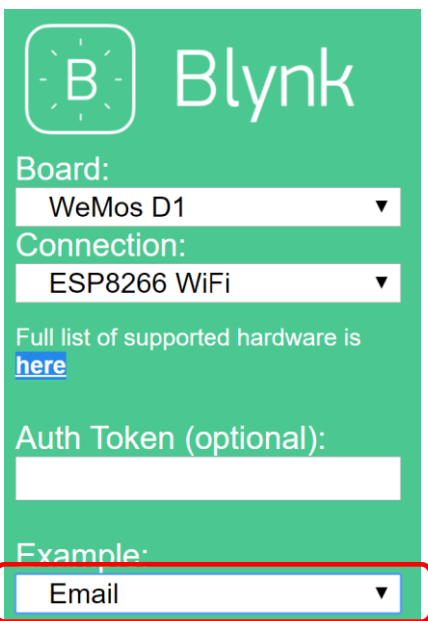
Assignment 3 below is also similar to that in **Arduino Lesson (2020) for WeMos D1**

- Hence, only extracts are shown.
- **Step 4** is different as it involves finding the sketch for email.

Assignment 3: Install Blynk library and sketch on Arduino IDE

3. Go to **Sketch** → **Include Library** menu → scroll down to check **Blynk** is present.
4. Go to website <http://help.blynk.cc/en/articles/512062-how-to-find-code-for-my-hardware> → **Open Blynk Examples Builder** to find sketch for email.
5. Copy this sketch into Arduino IDE.
6. Once **Auth Token** from Blynk app (**Assignment 4**) is available, update the sketch and upload into Arduino WeMos microcontroller.
7. Activate the **serial monitor** to show temperature and humidity readings via Blynk.

Note: In Arduino IDE, if you go to **File** > **Examples** > **Blynk** > **Widgets** > the sketch email – This uses Ethernet, **not** WIFI! `[#include <SPI.h> and #include <Ethernet.h>]`



Step 4

Reference:

- <https://www.geekstips.com/esp8266-email-and-push-notifications-iot-blynk/>

Assignment 3: Sample sketch

- In **blue**: original sketch for DHT11 data display with Blynk
- In **black**: additional lines of sketch for **email alert** via Blynk

```
/******  
Download latest Blynk library here: .....  
Downloads, docs, tutorials: http://www.blynk.cc  
Sketch generator: http://examples.blynk.cc  
Blynk community: http://community.blynk.cc  
*****  
This example shows how value can be pushed from Arduino to the Blynk App.  
WARNING :  
For this example you'll need Adafruit DHT sensor libraries:  
https://github.com/adafruit/Adafruit_Sensor  
https://github.com/adafruit/DHT-sensor-library  
  
App project setup:  
Value Display widget attached to V5  
Value Display widget attached to V6  
  
Blynk EMAIL and NOTIFICATION  
ESP8266 example  
www.geekstips.com  
*****/  
  
/* Comment this out to disable prints and save space */  
  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
#include <DHT.h>  
  
// You should get Auth Token in the Blynk App.  
// Go to the Project Settings (nut icon).  
char auth[] = "YourAuthToken"; // replace YourAuthToken with Auth Token  
  
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "YourNetworkName"; // replace YourNetworkName with Network SSID  
char pass[] = "YourPassword"; // replace YourPassword with Network key  
  
#define DHTPIN D4 // connect to digital pin 4  
  
#define DHTTYPE DHT11 // DHT 11  
  
DHT dht(DHTPIN, DHTTYPE);  
BlynkTimer timer;  
  
// This function sends Arduino's up time every second to Virtual Pin (5).  
// In the app, Widget's reading frequency should be set to PUSH. This means  
// that you define how often to send data to Blynk App.
```

```

void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // You can send any value at any time.
  // Please don't send more that 10 values per second.

  Serial.print("Humidity: ");      //show readings on serial monitor
  Serial.print(h);
  Serial.println(" %"); // println means start a new line after this
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C ");

  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);

  // SETUP the ALARM Trigger and Send EMAIL

  if (t > 30) {
    // Blynk API call for EMAIL sending with 3 parameters: email address, subject and body
    Blynk.email("your email address", "ESP8266 Alert", "Temperature over 30C!");

    // Blynk API call for NOTIFICATION pushing
    Blynk.notify("ESP8266 Alert - Temperature over 30 C!");
  }
}

void setup()
{
  // Debug console
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}


void loop()
{
  Blynk.run();
  timer.run();
}

```

Note:

- E.g. **Auth Token:** 275UBpwLoy3I5U1FdhvH0ZR6bsG4aVRx
 Network SSID: SINGTEL-1234
 Network Key: abcdefghij
- **Auth Token:** is obtained from **Blynk** email or Project Settings in **Blynk** app on smartphone
- **Network SSID** (Service Set ID) is the **network name** that displays when you refresh your wireless network list.

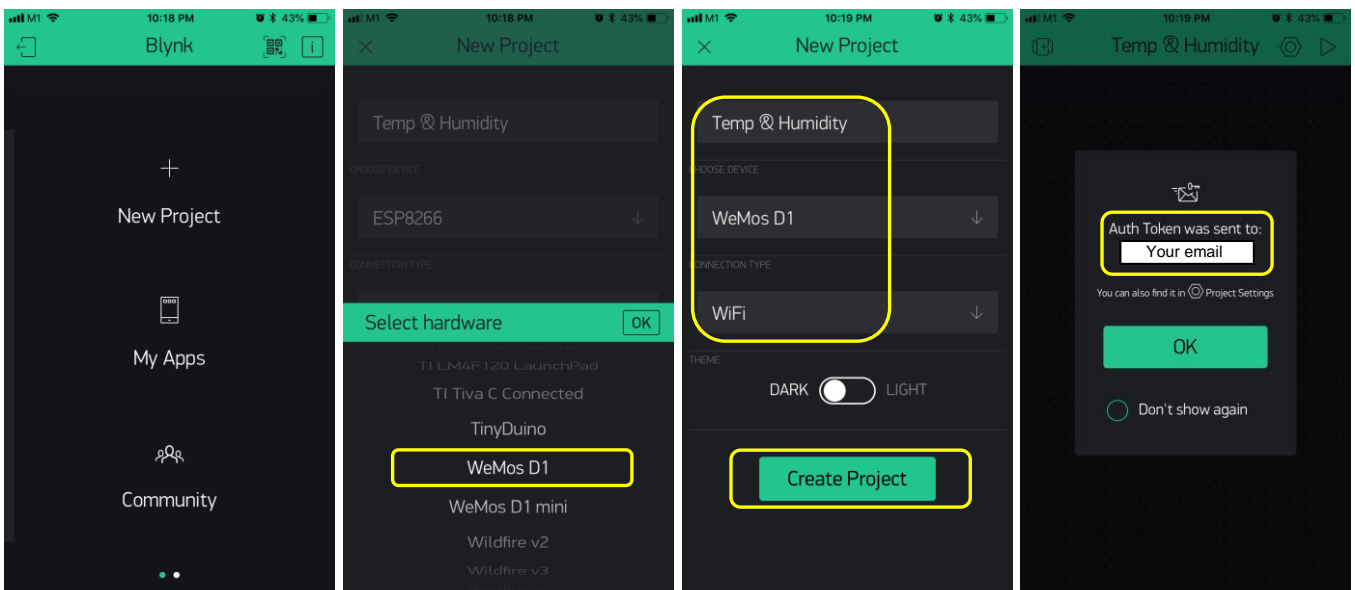
Assignment 4: Set up Blynk app on smartphone & temperature & humidity gauges

8. Download "**Blynk**" app from IOS Appstore.
 9. Sign up using your email (e.g. gmail – can reset password).
 10. Swipe sideways to "**New Project**". Enter your project name (if needed).
 - **Choose Device:** Scroll to select hardware "WeMos D1". Click OK.
 - **Connection Type:** Scroll to select connection type "WiFi".
 - Click **Create Project**. (After this, you will receive **Auth Token** in your email).
 - Click **OK**.
- Note:**
- Check email for **Auth Token**. (Also found in top right "nut icon"  **Project Settings**).
 - Insert **Auth Token** into the Blynk sketch in Assignment 3.
11. Tap on the empty canvas to open "**Widget Box**".
 12. Carry out the same steps as in previous **Arduino Lesson (2020)** for **WeMos D1** to create the 2 widgets for temperature and humidity using **Gauge** functions.
 13. Select to add "**email**" widget to the project.
 14. Under the project where you can see both gauges and email icon, click the **play button** or "**start**" **triangle** (top right corner) to activate both gauges. If wifi is connected, you would see the readings.
 15. Compare the readings on **Blynk** with those shown on **Serial Monitor**.
 16. Test the email widget by *increasing the temperature* of DHT11 sensor using a hairdryer!

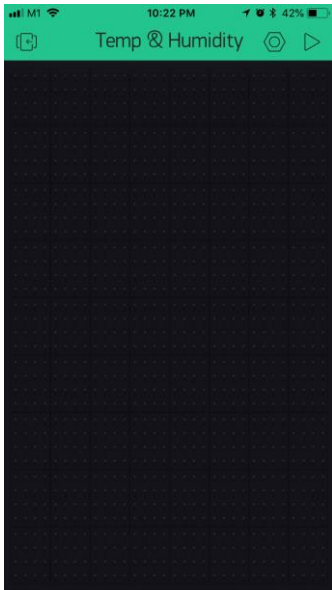
References:

- <https://docs.blynk.cc/>
- <http://help.blynk.cc/en/articles/580927-how-much-does-blynk-cost>
*In the Widget Box, there are limited "free" widgets to use (shown as Energy Balance). If the number below a widget is in **RED**, this is not longer free.*

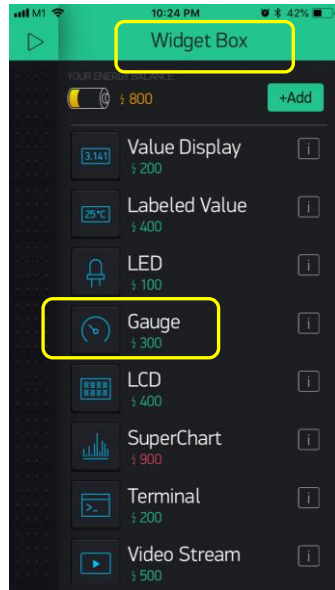
Screen shots of Blynk on smartphone



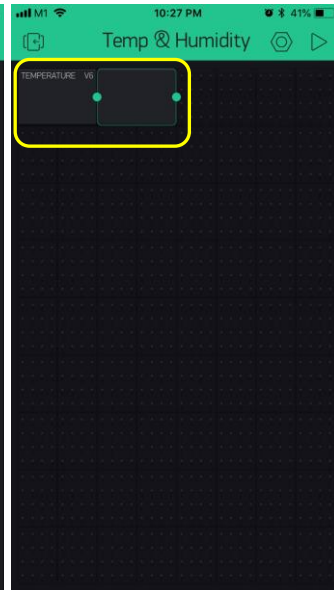
Step 10



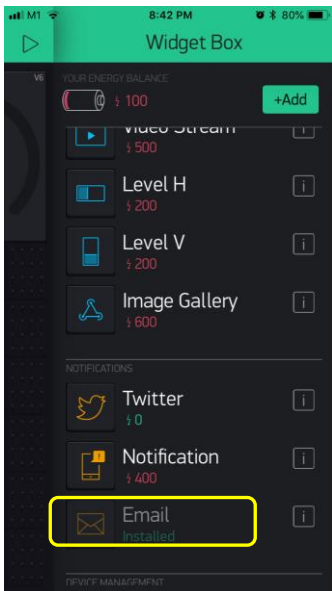
Step 11



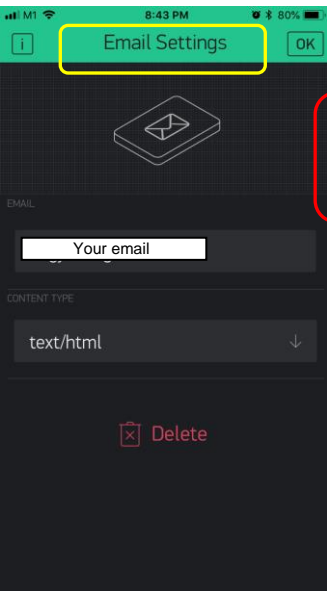
Step 12



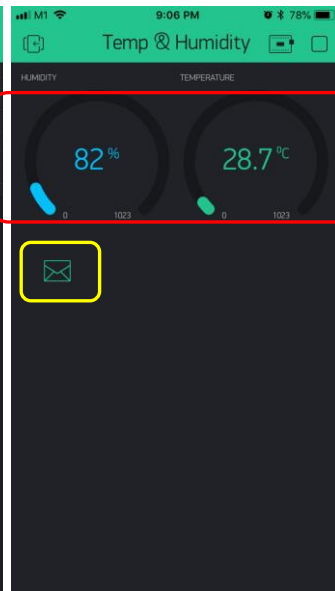
Step 12



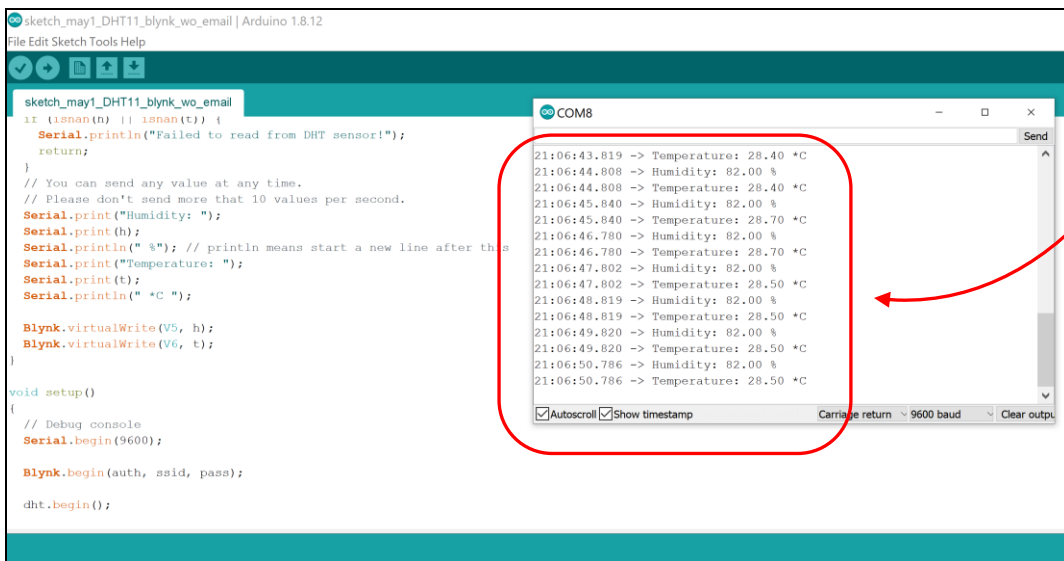
Step 13



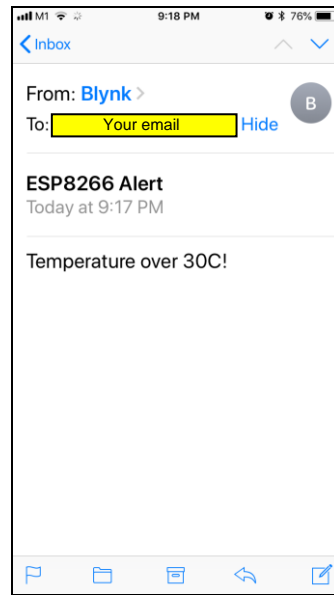
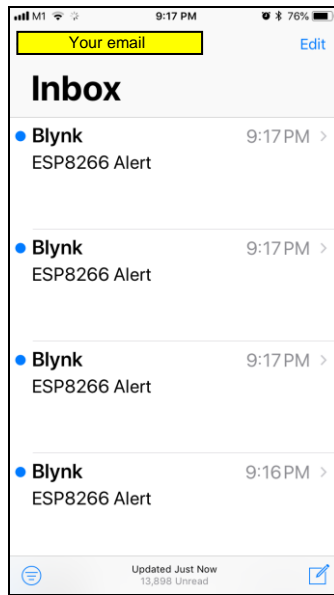
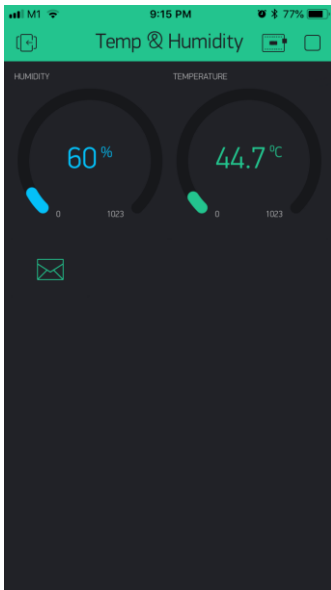
Step 13



Step 14



Step 15 (readings shown on **Serial Monitor**)



Step 16

Annex A

<p>From Blynk website (email widget) – using button Board: WeMos D1 Connection: ESP8266 WIFI Example: Email</p>	<p>From https://www.geekstips.com/esp8266-email-and-push-notifications-iot-blynk/ - Using DHT22 The Arduino example for a prototype using ESP8266 version 01 and DHT22 Temperature and Humidity sensor - hooked on the GPIO02 and defined as DHTPIN 2.</p>
<pre> ***** Simple e-mail example App project setup: E-mail Widget Connect a button to digital pin 2 and GND Pressing this button will send an e-mail WARNING: You are limited to send ONLY ONE E-MAIL PER 5 SECONDS! *****/ /* Comment this out to disable prints and save space */ #define BLYNK_PRINT Serial /* Set this to a bigger number, to enable sending longer messages */ //#define BLYNK_MAX_SENDBYTES 128 #include <ESP8266WiFi.h> #include <BlynkSimpleEsp8266.h> // You should get Auth Token in the Blynk App. // Go to the Project Settings (nut icon). char auth[] = "YourAuthToken"; // Your WiFi credentials. // Set password to "" for open networks. char ssid[] = "YourNetworkName"; char pass[] = "YourPassword"; unsigned count = 0; void emailOnButtonPress() { // *** WARNING: You are limited to send ONLY ONE E-MAIL PER 5 SECONDS! *** // Let's send an e-mail when you press the button // connected to digital pin 2 on your Arduino int isButtonPressed = !digitalRead(2); // Invert state, since button is "Active LOW" if (isButtonPressed) // You can write any condition to trigger e-mail </pre>	<pre> /* Blynk EMAIL and NOTIFICATION * ESP8266 example * www.geekstips.com */ #define BLYNK_PRINT Serial #include <ESP8266WiFi.h> #include <BlynkSimpleEsp8266.h> #include <SimpleTimer.h> #include <DHT.h> // You should get Auth Token in the Blynk App. // Go to the Project Settings (nut icon). char auth[] = "*****"; // Your WiFi credentials. // Set password to "" for open networks. char ssid[] = "*****"; char pass[] = "*****"; #define DHTPIN 2 // What digital pin we're connected to // Uncomment whatever type you're using! //#define DHTTYPE DHT11 // DHT 11 #define DHTTYPE DHT22 // DHT 22, AM2302, AM2321 //#define DHTTYPE DHT21 // DHT 21, AM2301 int alarmPin = 4; int led1 = 16; int led2 = 14; DHT dht(DHTPIN, DHTTYPE); SimpleTimer timer; void sendSensor(){ float h = dht.readHumidity(); float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit if (isnan(h) isnan(t)) { Serial.println("Failed to read from DHT sensor!"); return; } </pre>

<pre> sending { Serial.println("Button is pressed."); // This can be seen in the Serial Monitor count++; String body = String("You pushed the button ") + count + " times."; Blynk.email("your_email@mail.com", "Subject: Button Logger", body); // Or, if you want to use the email specified in the App (like for App Export): //Blynk.email("Subject: Button Logger", "You just pushed the button..."); } void setup() { // Debug console Serial.begin(9600); Blynk.begin(auth, ssid, pass); // You can also specify server: //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80); //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080); // Send e-mail when your hardware gets connected to Blynk Server // Just put the recipient's "e-mail address", "Subject" and the "message body" Blynk.email("your_email@mail.com", "Subject", "My Blynk project is online."); // Setting the button pinMode(2, INPUT_PULLUP); // Attach pin 2 interrupt to our handler attachInterrupt(digitalPinToInterrupt(2), emailOnButtonPress, CHANGE); } void loop() { Blynk.run(); } </pre>	<pre> Serial.println(t); Blynk.virtualWrite(V5, h); Blynk.virtualWrite(V6, t); // SETUP the ALARM Trigger and Send EMAIL // and PUSH Notification if (t > 28){ Blynk.email("vaduva.ionut.lucian@gmail.com", "ESP8266 Alert", "Temperature over 28C!"); Blynk.notify("ESP8266 Alert - Temperature over 28C!"); } } void setup(){ Serial.begin(9600); Blynk.begin(auth, ssid, pass); dht.begin(); timer.setInterval(2500L, sendSensor); } void loop(){ Blynk.run(); timer.run(); } </pre>
--	---

Note:

- <https://community.blynk.cc/t/the-difference-between-simpletimer-and-blynktimer/38934/2>
No need to install <Simpletimer.h> separately:

BlynkTimer allows you to send data periodically with given intervals not interfering with Blynk library routines. Blynk Timer inherits SimpleTimer Library, a well known and widely used library to time multiple events on hardware. BlynkTimer is included in Blynk library by default and there is no need to install SimpleTimer separately or include SimpleTimer.h

- <https://docs.blynk.cc/#widgets-notifications-email>

Example code

Limitations

- Maximum allowed email + subject + message length is 120 symbols. However you can increase this limit if necessary by adding #define BLYNK_MAX_SENDBYTES XXX to you sketch. Where XXX is desired max length of your email. For example for ESP you can set this to 1200 max length #define BLYNK_MAX_SENDBYTES 1200. The #define BLYNK_MAX_SENDBYTES 1200 must be included before any of the Blynk includes.
- Only 1 email per 5 seconds is allowed
- In case you are using gmail on the Local Server you are limited with 500 mails per day (by google). Other providers may have similar limitations, so please be careful.
- User is limited with 100 messages per day in the Blynk Cloud;