

## Programming ESP8266- ESP-201 stand alone with Arduino IDE

by [nataku](#) on November 12, 2015

### Table of Contents

Programming ESP8266- ESP-201 stand alone with Arduino IDE .....	1
Intro: Programming ESP8266- ESP-201 stand alone with Arduino IDE .....	2
Step 1: Setup your Arduino IDE .....	2
Step 2: Select the board .....	2
Step 3: Hook up your ESP8266. ....	3
Step 4: Blynk app/page setup .....	4
Step 5: Running the sketch in the ESP8266 ESP-201 .....	6
Step 6: Going further .....	6
File Downloads .....	7
Related Instructables .....	7
Advertisements .....	7
Comments .....	7

## Intro: Programming ESP8266- ESP-201 stand alone with Arduino IDE

I want to provide a better documentation for how you can use ESP201 (a variant of ESP8266) as stand alone board and how to hook up the GPIO in order to program it with the Arduino IDE.

I was struggling on how to get the sketch from Arduino IDE downloaded into the ESP8266 so I wanted to share what I found if there are folks out there that use ESP-201 as stand alone ESP8266 with Arduino IDE.

In this example I am using ESP-201 I bought from Ebay. It is the Ai-thinker version but I don't think it matters.

The other program I use is the BLYNK program.

<http://docs.blynk.cc/#downloads-blynk-apps-for-ios-or-android>

<http://docs.blynk.cc/#downloads-blynk-library>

I will let you read a bit more about BLYNK from their website.

I am using Arduino IDE 1.6.4.



## Step 1: Setup your Arduino IDE

I am using version 1.6.4,

You need to go to the Preference and add the following to the URLs for Boards Manager

[http://arduino.esp8266.com/stable/package\\_esp8266c...](http://arduino.esp8266.com/stable/package_esp8266c...) in order to add the ESP8266 board.

## Step 2: Select the board

Select the Generic ESP8266 board. (This is the one I am using, but perhaps if you use the other ESP8266 firmware you should try the other one). I didn't put any new firmware in mine. I just use it as it came from E-bay.

The rests of the setup under the board are left as default. I don't think they matter much, at least in my case I didn't have to change anything.



### Step 3: Hook up your ESP8266.

Okay now it is time to get your breadboard ready. You need to have 3.3V supply available and also a separate 5V supply. Your FT232RL break out board may have a 5V source you can use. I use FT232RL from Sparkfun.

You will also need to get your level converter, I use the bidirectional version from Adafruit.

Okay, now for the ESP-12 connection

1. Connect 3.3V source to the following pins : 3.3V, CHP\_Enable
2. Connect Gnd source to the following pins : Gnd, GPIO 00, GPIO 15.
3. Connect TX to the RX via the bidirectional logic level converter board. ESP-201 TX should go to the 3.3V side.
4. Connect RX to the TX via the bidirectional logic level converter board. ESP-201 RX should go to the 3.3V side.
5. Don't forget to connect the 3.3V source to power the LV side of the bidirectional logic level converter.
6. Lastly connect the ground from your 3.3V supply to the ground on the bidirectional logic level converter.

Then from your FT232RL, use the 5V source on the board to power the HV side of the bidirectional logic level converter board, and take the corresponding high-side of the ESP-201 TX to the RX and do the same for the ESP-201 RX into the TX of the FT232RL. And of course, don't forget the ground from FT232RL break out board to the logic level converter board.

You should go to the BLYNK website and download their Arduino Library then load their ESP8266 stand alone example. Fill out the auth token and your wifi id and password. You will get the auth token by making a page on your BLYNK App. Let us just put a simple push button, with the output set to GP14 (refer to GPIO14 in ESP201).

You don't have to write any code to program a push button in the Arduino sketch for this first easy example.

Okay BEFORE You HIT DOWNLOAD.... do this

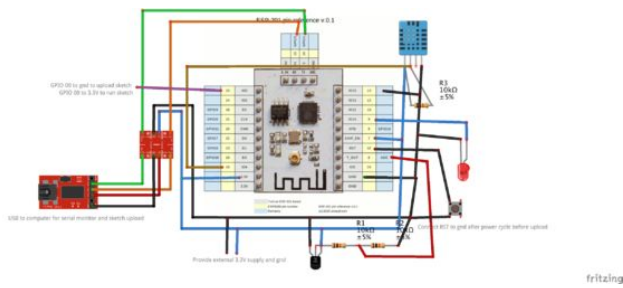
1. Power down the 3.3V source and turn it back on, and connect the RST pin of the ESP-201 to the ground source momentarily (ie. pulse). This I think put the ESP-12 into boot mode.
2. Then HIT the COMPILE and DOWNLOAD.

If you keep getting the message saying esp sync not responding, like this

```
warning: espcomm_sync failed error: espcomm_open failed
```

it means you did not get the ESP8266 into boot mode and recheck your GPIO and the wiring setup and make sure you see my note on Cycling the power, and hitting the RST pin momentarily. This is the missing part from a lot of the other instructions.

Anyway, if everything goes well, you should see the note saying uploading bla...bla...bla.... and a bunch of dots..... and finally it will say Done Uploading.



#### Step 4: Blynk app/page setup

Well, I was going to write a bit more detail here on how to design the Blynk Page for going a bit further than just a push button but may be I'll wait until I write the next step and see.

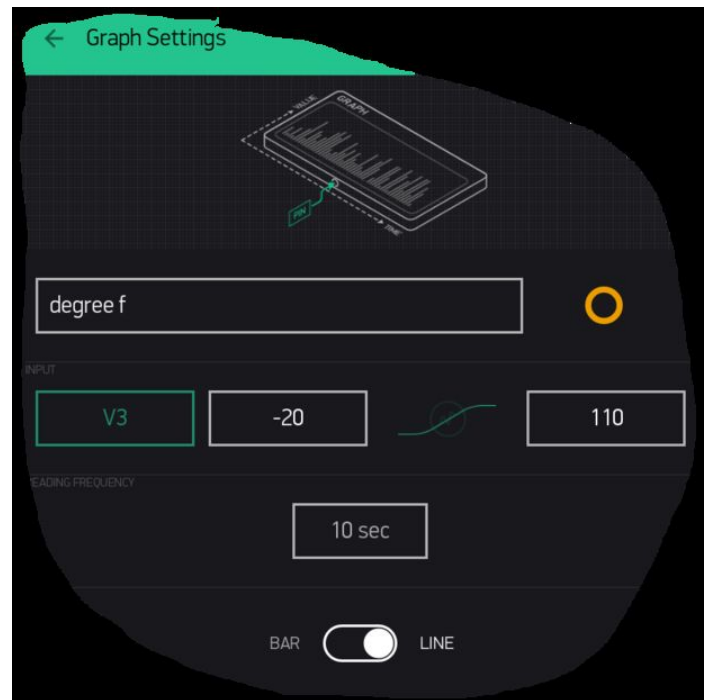
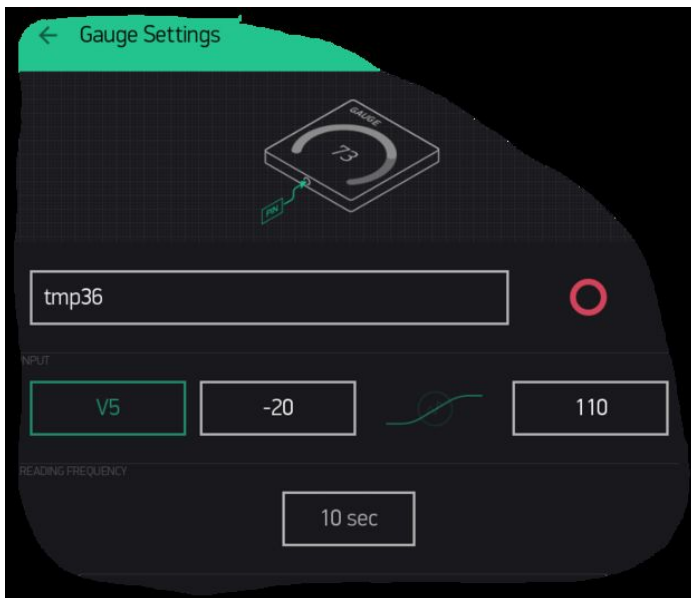
So the Blynk page setup here started with a simple push button that turn on/off an LED.

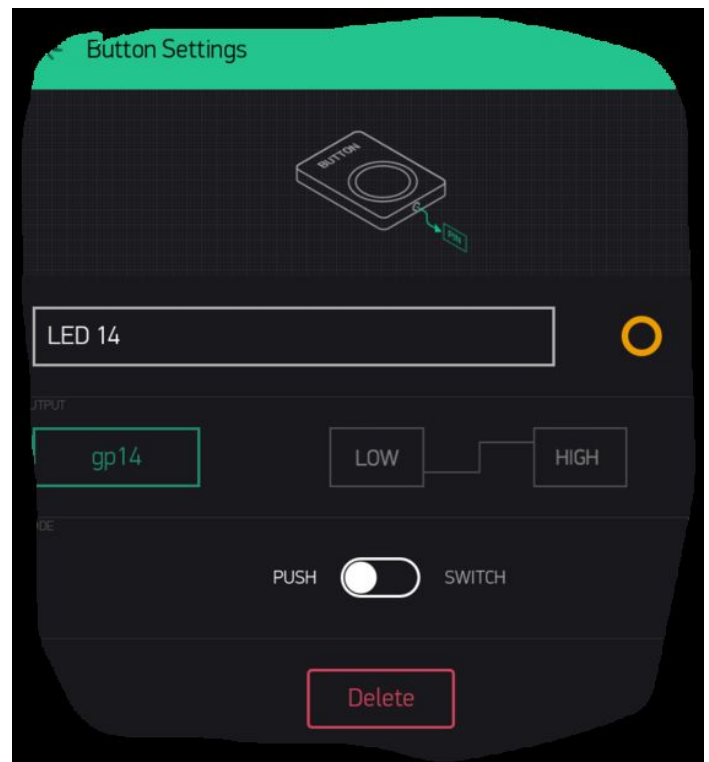
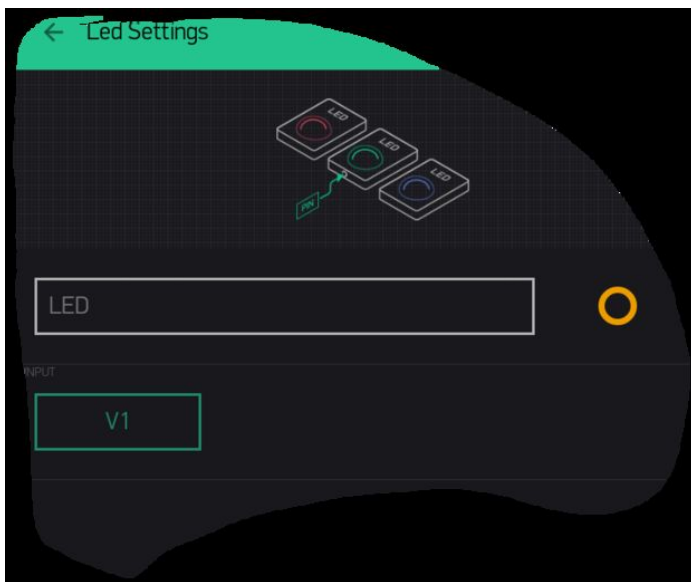
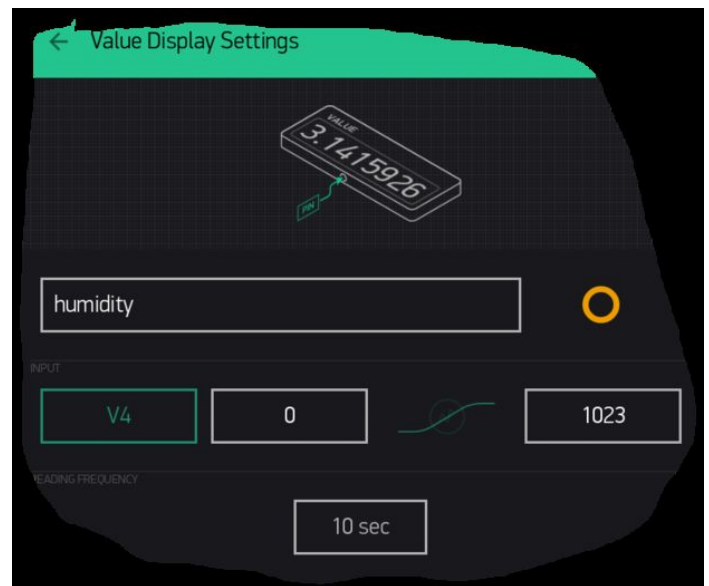
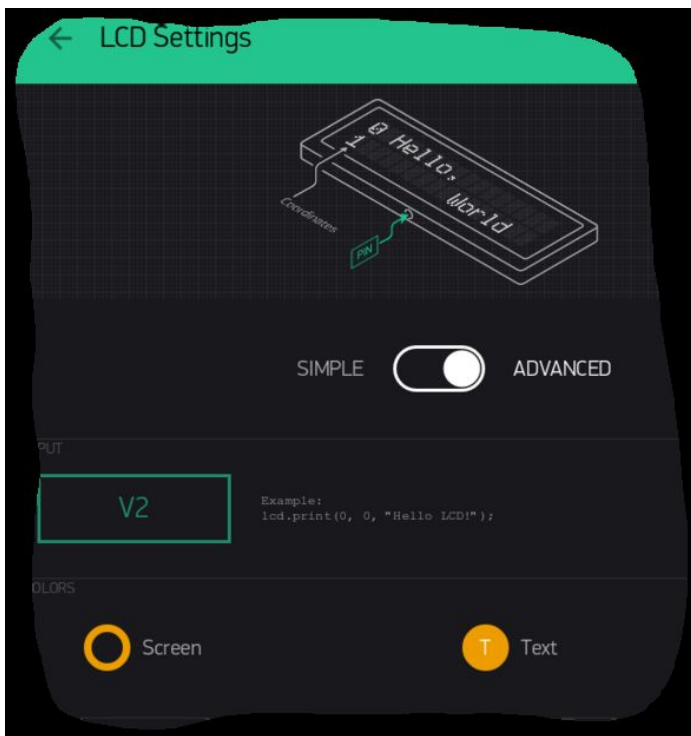
Then I try to show how you can read DHT11 and TMP36 data and display it on the BLYNK page.

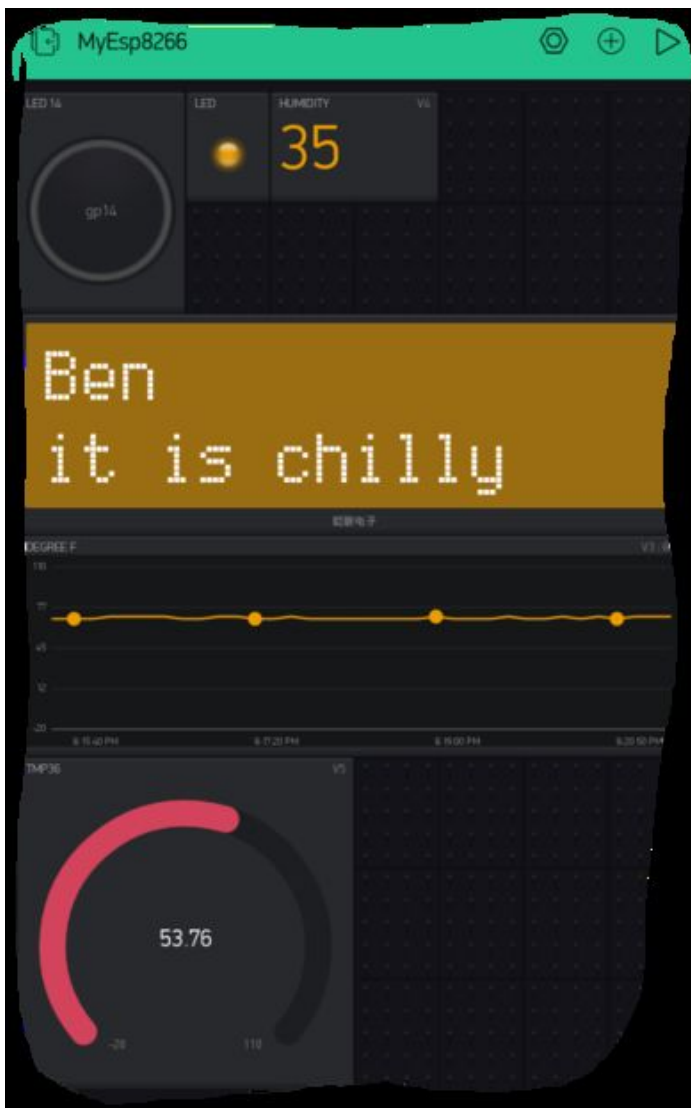
I also used the Gauge, Values and Graph widget. They are really nice.

I also automatically toggle an LEDWidget on the Blynk Page to turn on and off every 2 second.

There is also an LCD that you can display pretty much anything but I just write some simple if and then function to display preset text based on the temperature.







### Step 5: Running the sketch in the ESP8266 ESP-201

Once you finish uploading the sketch, you should be able to go to your tablet/phone where you have your BLYNK page/controller setup and hit the LED button and turn on the LED.

I found that once you download the IDE, the sketch will run and you can use serial monitor to check if you are connecting to the Blynk server.

However, if you turn off the power and turn it back on, it won't connect to your wifi as if your sketch is not running.

So you need to do these.

1. Make sure that GPIO 00 is no longer connected to ground.
2. Keep GPIO 15 connected to ground.
3. Cycle the power again on the ESP-201 and watch your serial monitor. It should indicate you are connecting to your wifi SSID. Then you should connect right away to your wifi and eventually to the BLYNK server.

### Step 6: Going further

So let's try adding a few functions to learn how the GPIO numbers relate or are used by the BLYNK library.

Simple GPIO on/off controller

As far as I can tell the GPIO printed on the ESP-201 are exactly the BLYNK modules use. So for example if you are programming or creating a BLYNK MODULE with a push button controlled by GPIO-14, this is referred to as IO 14 in the ESP-201. You can then wire an LED or relay control on the pin.

Simple Digital data read from sensor

Let's take another example, let's say you are going to wire in a DHT11 sensor into GPIO-04 / IO 04. You will refer to this as pin #4 in your sketch/program. A little note on using the DHT11, I am using the DHT.h library from Adafruit. If you look at their wonderful website, they indicate to pass a third parameter when using DHTxx type sensor with ESP8266 because it is a faster CPU. So don't forget that. (for example `DHT dht(DHTPIN,DHTTYPE,11)` )

The number 11 I don't think relates to the type of DHT but not too sure so if you use DHT22, you can still use the number 11.

Simple Analog Read

<http://www.instructables.com/id/Programming-ESP8266-ESP-201-stand-alone-with-Arduino/>

How about TMP36, well this is a bit tricky but if you search the internet, you will find lots of people already try this. The important part is the ADC (Analog to Digital input) is referred to as Pin 17. It is always 17 if you are using ESP8266. This is coded in one of the library file for ESP8266. I don't remember which one.

And the other thing to remember is the reference voltage is around 1.10 Volt. And there is a maximum voltage limit to this ADC pin. So don't just let 3.3V go all the way in. Use a voltage divider and adjust your conversion table or formula in your code accordingly with this voltage divider. In my sketch example, I use equal value of R1 and R2 so the math is simple (a factor of 2).

#### Simple Timer

I think another important one here is usage of BLYNK SimpleTimer library. They recommend using this as a way to do other function so there is no usage of "delay()" function inside the loop() as this can disconnect your connection. It is pretty easy to use, essentially just define the period of time you want the function to be done like read sensor or toggle led, etc. in your setup(). Then you write the read sensor function as another call function outside the loop.

There are more information on the BLYNK website.

Hopefully this helps you out.

#### File Downloads



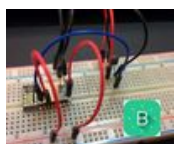
**MyESP8266\_Standalone\_blynk\_test\_v2.ino** (3 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'MyESP8266\_Standalone\_blynk\_test\_v2.ino']

#### Related Instructables



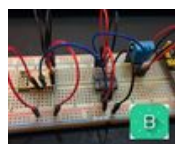
**ESP8266-12  
standalone  
blynk control  
outlets wireless  
433 mhz** by  
catoplepa



**ESP8266-12  
blynk wireless  
temperature  
LM35 sensor** by  
catoplepa



**ESP8266 ESP-  
12(Standalone)+  
Blynk 101** by  
Rahul27



**ESP8266-12  
blynk wireless  
temperature,  
humidity DHT  
11 sensor** by  
catoplepa



**ESP8266-12  
blynk wireless  
simple virtual  
led** by catoplepa



**ESP8266-12  
blynk wireless  
temperature,  
humidity DHT22  
sensor** by  
catoplepa

#### Comments